Application des réseaux de tenseurs pour l'étude du bruit avec mémoire dans les codes polaires et ses généralisations

par

Benjamin Bourassa

Mémoire présenté au département de physique

en vue de l'obtention du grade de maître ès sciences (M.Sc.)

FACULTÉ des SCIENCES

UNIVERSITÉ de SHERBROOKE

Sherbrooke, Québec, Canada, 22 janvier 2019

Le 22 janvier 2019

le jury a accepté le mémoire de Monsieur Benjamin Bourassa dans sa version finale.

Membres du jury

Professeur David Poulin
Directeur de recherche
Département de physique

Professeur Glen Evenbly
Rapporteur
Département de physique

Professeur Ion Garate
Membre
Département de physique

J’aimerais dédier ce mémoire à ma grand-mère,

Sommaire

Les codes polaires sont une famille de codes de correction d'erreurs récemment développées. Leurs bonnes propriétés font en sorte qu'ils ont été sélectionnés pour être intégrés dans les nouveaux standards de télécommunication (5G) vers 2020. L'algorithme de décodage, connu sous le nom de décodeur par annulation successive, est une méthode de décodage itérative. Cet algorithme permet aux codes polaires d'atteindre le taux de transmission d'informations optimal qu'on peut transmettre pour un modèle de bruit donné.

Ce mémoire présente une méthode de décodage graphique basée sur les réseaux de tenseurs, un outil très utilisé en physique des problèmes à N-corps. L'aspect innovant de ce mémoire est la prise en compte d'un modèle de bruit avec mémoire, une généralisation du décodeur par annulation successive. Il est possible d'étendre la famille des codes polaires en introduisant une structure interne convolutive. Ces codes ainsi obtenus sont nommés les codes polaires convolutifs. L'algorithme de décodage reste valide pour ces généralisations.

Les résultats, obtenus par simulation numérique, montrent que la performance des codes polaires convolutifs est supérieure à celle des codes polaires standard pour les modèles de bruits considérés.

Mots-clés : Correction d'erreurs, codes polaires, codes polaires convolutifs, réseaux de tenseurs, canaux à états finis.

Remerciements

La réalisation de ce mémoire n'aurait été possible sans certaines personnes que je tiens à remercier. Je commence par mon directeur de maîtrise David Poulin. Merci de m'avoir accompagné dans l'exécution de mon projet de maîtrise. Au sein de ton groupe, j'ai eu l'occasion de participer à plusieurs écoles et conférences internationales, ces expériences furent très enrichissantes.

Je remercie Jessica pour avoir été une collègue de bureau exemplaire. J'ai eu beaucoup de plaisir à tes côtés au cours des deux dernières années. J'ai apprécié ta grande écoute durant mes interrogations ainsi que ton amitié.

Je remercie Maxime d'avoir été un coéquipier de maîtrise incroyable. Ta joie de vivre et ta confiance m'ont aidé à propulser ma maîtrise.

Je remercie aussi l'ensemble des membres du groupe de David Poulin pour l'aide et les dicussions enrichissantes. Merci spécialement à Pavi pour ses conseils.

Je remercie Lucas pour avoir été un coloc formidable. Ton écoute et tes conseils judicieux m'ont aidé à travers mon parcours de maîtrise et de vie.

Je remercie Mathieu pour m'avoir permis de rentrer dans ton bureau à tout moment pour parler de tout et de rien.

Je remercie la gang du soccer pour nos matchs incroyables !

J'aimerais aussi remercier mes amis Léo, Martin, Félix-Antoine et Félix alias le gars des signets.

Merci à ma famille pour votre soutien dans mes études. Merci maman et papa

de m'avoir toujours encouragé dans la poursuite de mes objectifs. Finalement, merci à Lisa pour ton écoute et ton amour.

Table des matières

Liste des tableaux

Table des figures

Chapitre 1

Introduction

Une des plus grandes révolutions du 20^{e} siècle est sans aucun doute l'invention du transistor qui mènera quelques années plus tard à l'ère du digital avec ces technologies qui nous sont bien familières aujourd'hui. À cette même époque, un jeune scientifique nommé Claude Shannon fut une découverte tout aussi impressionnante, mais moins connue peut-être pour son caractère plus fondamental. Cette découverte est une théorie mathématique de la communication, elle permit entre autres la naissance de la théorie de l'information avec d'importantes applications en transmission de signaux numériques. Les découvertes de Shannon ont permis de donner une description mathématique de l'information. L'information au sens large semble difficile à quantifier mathématiquement puisqu'elle est intangible et abstraite, mais Shannon parvint à en donner une définition adéquate.

L'ensemble de ce mémoire s'intéresse au problème de communication, il s'agit d'une partie de l'ensemble des travaux de Shannon. Le problème de communication est le suivant : supposons deux parties *Alice* et *Bob* qui souhaitent communiquer de manière efficace en présence de bruit. Par exemple, Alice pourrait envoyer à Bob un message m. Ce message doit être transmis sur un canal bruyant qui introduira des erreurs au message. Comment Bob peut-il retrouver m avec forte probabilité ? Une manière de résoudre ce problème consiste à ajouter de la redondance au message de manière à le rendre robuste aux erreurs. Ainsi, Alice encodera son message m dans une chaîne contenant de la redondance x. Ensuite, x sera transmis sur le canal produisant une chaîne y. De son côté, Bob observera seulement la chaîne reçue y. Sa

FIGURE 1.1 Schéma d'une procédure de communication entre 2 parties implémentant un code de correction d'erreurs.

tâche est donc d'inférer le message d'Alice m sachant la chaîne reçue y. Il s'agit de la procédure de décodage. Ce principe est à la base de la correction d'erreurs. La figure 1.1 illustre un schéma simplifié du protocole de communication implémentant un code de correction d'erreurs.

La façon la plus simple d'ajouter de la redondance est d'utiliser la répétition. Par exemple, si Alice souhaite envoyer le bit 0 à Bob, elle pourrait envoyer 3 copies de ce bit, soit la chaîne 000. Supposons que le canal produit des erreurs de renversement de bits. C'est-à-dire qu'une erreur sur le bit 0 produit un 1 et vice-versa. Ainsi, une erreur peut survenir sur un des 3 bits et Bob peut décoder le message en utilisant le vote par majorité. En effet, une erreur produit l'une des 3 chaînes suivantes : 100, 010, 001 $\longrightarrow$ 0. Par contre, si 2 erreurs surviennent, le décodage par vote majoritaire ne permet pas une correction adéquate. Les 3 chaînes d'erreurs possibles sont : 011, 101, 110 $\longrightarrow$ 1. Le code à répétition permet donc l'envoi d'un bit d'information en utilisant 3 bits nommés bits physiques et il peut tolérer au plus 1 seule erreur sur les bits physiques. La performance d'un code de correction d'erreurs s'évalue grâce au *rendement d'un code* et à la *probabilité d'erreur après décodage*. Premièrement, le taux de transmission ou le rendement d'un code R se définit comme étant la fraction de bit d'information envoyée dans un message sur le nombre de bits physiques utilisés. Dans le cas du code à répétition, le rendement est $R = \frac{1}{3}$. Deuxièmement, la probabilité d'erreur après décodage P_L est une quantité reliée au nombre d'erreurs que peut tolérer un code de correction d'erreurs. Par exemple, pour un canal avec une probabilité d'erreur p, il faut faire au minimum 2 erreurs dans le code à répétition pour avoir une erreur logique de l'ordre de $P_L \sim p^2$. Le code de répétition offre donc une robustesse au niveau du message envoyé car $p^2 < p$. Par contre, il faut payer un coût en information de 3 bits envoyés pour 1 bit d'information. De manière générale, ces deux paramètres sont optimisés de la manière à avoir un rendement le plus près de 1 possible tout en ayant une probabilité d'erreur après décodage très faible. Un des succès de Shannon a été de montrer qu'il est possible d'atteindre ces conditions de manière optimale en fonction d'un modèle de bruit. Pour ce faire,

il suffit de choisir le bon code de correction d'erreurs. Il s'avère que cette tâche est extrêmement complexe. Les codes qui satisfont les conditions de Shannon se nomment les codes qui atteignent la capacité. En principe, il est possible d'utiliser une méthode de décodage dite optimale. Toutefois, celle-ci nécessite une complexité de décodage qui grandit de manière exponentielle avec la quantité d'information envoyée. Cette notion de complexité de décodage est importante à considérer puisqu'elle est intimement liée au temps d'exécution et au coût énergétique que pourrait avoir l'implémentation d'un tel code dans une technologie de la vie courante.

Depuis ce résultat de Shannon prouvé dans les années 40, les scientifiques ont tenté de trouver des codes qui atteignent la capacité, mais c'est seulement en 2009 avec l'invention des codes polaires par E. Arikan qu'un premier exemple réalisable de manière efficace est donné. En effet, la complexité d'encodage et de décodage est de $O(N \log N)$ où N est la taille du code. Ces codes prometteurs ont même été annoncés comme une des technologies utilisées dans la cinquième génération de standards pour la téléphonie mobile (5G) [6]. Le but de mon projet de maîtrise est l'étude des codes polaires et ses généralisations dans un contexte de bruit avec mémoire, c'est-à-dire, un bruit dont les erreurs sont corrélées. Ce type de modèle de bruit est intéressant puisqu'il permet de simuler bon nombre de processus causant dans erreurs dans des applications technologiques réelles. Par exemple, les erreurs causées par une rayure ou de la poussière sur un disque compact sont des erreurs corrélées. Ainsi, des méthodes de correction d'erreurs sont appliquées pour traiter l'information sur un disque compact offrant une résilience aux erreurs corrélées. Dans ce projet, les réseaux de tenseurs, un outil graphique populaire en physique des systèmes quantiques à N-corps, sont utilisés afin de reformuler l'algorithme de décodage standard des codes polaires. Cette formulation facilite l'implémentation d'algorithmes et permet de généraliser les codes polaires en les codes polaires convolutifs, une famille de codes introduite antérieurement par A. Ferris et D. Poulin. L'aspect innovant de se mémoire provient d'une généralisation du décodeur pour les codes polaires et les codes polaires convolutifs permettant de prendre en compte les canaux à états finis, un type de modèle de bruit avec mémoire. Les résultats de simulation obtenus montrent que ce décodeur surpasse les performances d'un décodeur qui ne prend pas en compte une structure de bruit avec mémoire. Une comparaison avec les codes polaires et les codes polaires convolutifs dans le contexte du

bruit avec mémoire indique que les codes polaires convolutifs offrent de meilleures performances que les codes polaires.

Le chapitre 2 de ce mémoire traite des fondements théoriques du projet en introduisant les notions de base en théorie des codes et en théorie de l'information. Les codes polaires sont ensuite présentés en spécifiant la récente découverte dans le cas du bruit avec mémoire. Cela est suivi d'une introduction aux réseaux de tenseurs pour finalement traiter des codes polaires convolutifs. Le chapitre 3 présente les contributions majeures provenant du projet de maîtrise. Il est question de la correspondance entre le problème du décodage et la contraction d'un réseau de tenseurs. Ensuite, le décodeur par annulation successive, appliqué au cas des codes polaires et des codes polaires convolutifs, est adapté au modèle de bruit avec mémoire grâce à l'usage des réseaux de tenseurs. Le chapitre 4 présente les résultats obtenus par simulation numérique suivie d'une analyse. Finalement, le chapitre 5 conclut et offre les objets de recherche future.

Chapitre 2

Théorie et mise en contexte

2.1 Théorie des codes

Pour caractériser un code de correction d'erreurs, il est important de parler du nombre de bits d'information k, du nombre de bits physiques n et de la distance d'un code d. Typiquement, un code est dénoté par ces 3 paramètres de la manière suivante : $[n, k, d]$. Par exemple, le code à répétition à 3 bits est donné par $[3, 1, 3]$. Le rendement d'un code est défini par $R = \frac{k}{n}$. Cette quantité correspond à la fraction d'un bit d'information transmis à chaque utilisation du canal. La plupart des codes de corrections d'erreurs utilisés dans les applications de la vie courante font partie de la famille des codes linéaires. Mathématiquement, la théorie des corps finis est utilisée pour étudier les codes de corrections d'erreurs [1]. Ainsi, un message $\vec{m}$ [2] de k bits est un élément de $\mathbb{F}_2^k$, l'espace vectoriel de dimension k à coefficients binaires et muni de l'addition modulo 2 dénotée par le symbole $\oplus$.

1. Spécifiquement dans ce mémoire, il s'agira du corps à deux éléments $\mathbb{F}_2$.
2. Dans cette section, sauf indication contraire, tous les vecteurs sont des vecteurs lignes.

2.1.1 Les codes linéaires

Dans cette section, la notion d'un code de correction d'erreurs est introduite selon 3 définitions équivalentes, traitant des concepts d'*espace vectoriel*, de *matrice génératrice* et de *matrice de contrôle*. Un code de correction d'erreurs comprend un encodeur $\mathscr{E}$, une application linéaire et injective, qui prend en entrée une chaîne de bits de message $\vec{m} \in \mathbb{F}_2^k$ et produit un mot de code $\vec{x} \in \mathbb{F}_2^n$ avec $k < n$. L'injectivité assure l'obtention de 2^k mots de code distincts provenant des 2^k chaînes de bits de message.

Définition 1 *Un code de correction d'erreurs est défini comme un sous-espace vectoriel de $\mathbb{F}_2^n$ ayant une dimension k,*

$$\mathcal{C} = \{\vec{x} \in \mathbb{F}_2^n | \ \vec{x} = \mathscr{E}(\vec{m}) \ , \ \vec{m} \in \mathbb{F}_2^k\}.$$

Une conséquence de la linéarité est que si $\vec{x_1}$ et $\vec{x_2} \in \mathcal{C}$, alors $\vec{x_1} \oplus \vec{x_2} \in \mathcal{C}$. Il est possible de spécifier une base pour le sous-espace vectoriel $\mathcal{C}$. Cette base contient k vecteurs de n bits qui sont appelés les générateurs $\{\vec{g_1}, \vec{g_2}, ..., \vec{g_k}\}$. Ainsi, tout mot de code $\vec{x} \in \mathcal{C}$ peut être représenté comme une combinaison linaire des vecteurs de bases : $\vec{x} = \sum_{i=1}^{k} a_i \vec{g_i}$. Il est possible de définir une matrice G de dimension $k \times n$ dont chaque ligne est formée d'un générateur. Cette matrice est appelée la matrice génératrice et elle définit l'encodage de la manière suivante : $\vec{x} = \vec{m}G$. Ceci donne donc une deuxième définition d'un code de correction d'erreurs prenant en compte la spécificité d'un code linaire.

Définition 2 *Un code de correction d'erreurs est défini par une matrice génératrice de dimension $k \times n$,*

$$\mathcal{C} = \{\vec{x} \in \mathbb{F}_2^n | \ \vec{x} = \vec{m}G \ , \ \vec{m} \in \mathbb{F}_2^k\}.$$

Finalement, la matrice de contrôle H permet de donner des équations de contraintes sur les mots de codes. Selon la définition 1, un code est défini par un sous-espace vectoriel de dimension k sur $\mathbb{F}_2^n$. En spécifiant les générateurs, le code $\mathcal{C}$ peut se définir par une matrice génératrice G. Il est donc possible de considérer l'espace vectoriel perpendiculaire à $\mathcal{C}$ noté $\mathcal{C}^{\perp}$.

Définition 3 *Le code dual $\mathcal{C}^\perp$ se définit comme l'espace vectoriel perpendiculaire au code $\mathcal{C}$,*

$$\mathcal{C}^\perp = \{\vec{c} \in \mathbb{F}_2^n |\ \langle \vec{c}, \vec{x} \rangle = 0\ \forall \vec{x} \in \mathcal{C}\}.$$

où $\langle \vec{c}, \vec{x} \rangle = \sum_{i=1}^{n} c_i x_i$.

Cet espace définit aussi un code qui encode $n-k$ bits d'information dans n bits. Posons H la matrice génératrice de $\mathcal{C}^\perp$ qui est de dimension $(n-k) \times n$. Ce code est le code dual de $\mathcal{C}$ et donc par définition $HG^T = 0$. Supposons donc un code $\mathcal{C}$ avec la matrice génératrice G et H la matrice génératrice de $\mathcal{C}^\perp$. Si $\vec{x} \in \mathcal{C}$ obtenus par $\vec{x} = \vec{m}G$, alors $H\vec{x}^T = HG^T\vec{m}^T = 0$. Ceci mène donc à une autre définition de $\mathcal{C}$:

Définition 4 *Un code de correction d'erreurs est défini par une matrice de contrôle de dimension $(n-k) \times n$,*

$$\mathcal{C} = \{\vec{x} \in \mathbb{F}_2^n |\ H\vec{x}^T = 0\}.$$

Cette définition est très utile puisqu'elle impose un ensemble de conditions sous la forme d'équations linéaires sur tous les mots de code indépendamment de la chaîne de bits d'information envoyée. Sans aucune notion supplémentaire, la condition $H\vec{x}^T = 0$ permet une manière simple de détecter la présence d'erreurs. En effet, soit $\vec{\tilde{x}}$ une chaîne de bits contenant des erreurs. Alors, par définition $\vec{\tilde{x}}$ n'est pas un mot de code de $\mathcal{C}$ et donc $H\vec{\tilde{x}}^T \neq 0$.

Comment effectuer la correction des erreurs ? C'est-à-dire, étant donné une matrice G ou H définissant un code et une chaine de bits possiblement erronée $\vec{y}$, est-il possible de trouver le mot de code $\vec{x}$ le plus probable ? Pour ce faire, il est utile de définir la notion de *distance de Hamming* entre deux chaînes de bits $\vec{u}$ et $\vec{v}$ de même longueur.

Définition 5 *La distance de Hamming entre deux chaînes de bits $\vec{u}$ et $\vec{v}$ de même longueur, notée $d_H(\vec{u}, \vec{v})$, est égale au nombre de positions où les éléments de $\vec{u}$ diffèrent avec les éléments de $\vec{v}$.*

$$d_H(\vec{u}, \vec{v}) = |\{i \mid u_i \neq v_i\}|$$

Définition 6 *Le poids de Hamming d'une chaîne $\vec{u}$ est $w_H(\vec{u}) = d_H(\vec{u}, \vec{0})$.*

Par exemple,

$$d_H(101011,\ 110100) = 5,$$

$$w_H(101011) = 4.$$

Soit un modèle de bruit agissant sur les différents mots de code par des erreurs linéaires [3], indépendantes et de faibles poids. Ce modèle de bruit est nommé le canal binaire symétrique et produit des erreurs d'inversion de bits avec une certaine probabilité. Soit un mot de code $\vec{x} \in \mathcal{C}$ et une erreur $\vec{e} \in \mathbb{F}_2^n$ telle que $w_H(\vec{e}) \leq s$ avec s un entier assez faible. Alors, la chaîne à la sortie du canal est obtenue par $\vec{y} = \vec{x} \oplus \vec{e}$. Le travail du décodeur consiste donc à trouver $\vec{x}$ avec la connaissance du code $\mathcal{C}$ et du vecteur $\vec{y}$. Une stratégie optimale est d'utiliser un *décodeur à distance minimale*, car si une erreur survient elle est de faible poids. Alors, le décodeur peut répertorier l'ensemble des 2^k mots de code et calculer la distance de Hamming entre tous ces mots de code et la chaîne de bits reçue pour retenir le mot de code ayant la distance de Hamming minimale. Mathématiquement, le décodeur à distance minimale cherche donc à optimiser la fonction suivante,

$$\vec{\tilde{x}} = \min_{\vec{x} \in \mathcal{C}} d_H(\vec{x}, \vec{y}). \tag{2.1}$$

Cette stratégie est optimale du fait que la probabilité d'avoir une erreur est une fonction décroissante de son poids. Par contre, cette méthode requiert une quantité énorme de calculs pour des codes très grands. De façon générale, la complexité de décodage est exponentielle selon k puisqu'il est nécessaire de parcourir l'ensemble des 2^k mots de code. La notion de distance de Hamming permet aussi de caractériser le paramètre de *distance d'un code*. Il s'agit d'un paramètre très important pour quantifier le poids maximal d'une erreur qu'un code peut tolérer.

3. Une erreur $\vec{e}$ agit linéairement sur un mot de code $\vec{x}$ si la chaîne obtenue à la sortie du canal est donnée par $\vec{z} = \vec{e} \oplus \vec{x}$.

Définition 7 *La distance minimale d'un code est donnée par*

$$d = \min_{\vec{x} \in \mathcal{C} \setminus \vec{0}} w_H(\vec{x})$$

le poids de Hamming minimal d'un mot de code non trivial.

Par exemple, pour le code à répétition à 3 bits, les deux mots de code sont $\vec{x}_0 = 000$ et $\vec{x}_1 = 111$, ce qui donne $d = 3$.

Théorème 1 *Un code de distance d peut détecter des erreurs de poids $\leq d - 1$ et corriger des erreurs de poids $\leq \lfloor \frac{d-1}{2} \rfloor$.*

Selon ce théorème, le code à répétition à 3 bits permet de détecter des erreurs de poids jusqu'à 2 et de corriger des erreurs de poids 1.

Une des méthodes typiques de décodage pour des codes de faibles tailles consiste à utiliser un décodage *par syndromes*. Soit un mot de code $\vec{z}$, alors le message reçu à la suite de la transmission sur un canal est donné par $\vec{r} = \vec{z} \oplus \vec{e}$ où $\vec{e}$ est l'erreur. Par définition, $\vec{z}$ est un mot de code et donc $H\vec{z} = 0$. Par linéarité,

$$\vec{s} = H\vec{r} = H(\vec{z} \oplus \vec{e}) = H\vec{z} \oplus H\vec{e} = H\vec{e} \tag{2.2}$$

où $\vec{s}$ est un vecteur de dimension $n-k$ appelé le syndrome de l'erreur $\vec{e}$. Cette quantité est intéressante puisqu'elle ne dépend pas du message envoyé, seulement de l'erreur. Avec cela, il est possible d'implémenter une table de syndromes qui associe à chaque vecteur d'erreurs $\vec{e}$ un syndrome. Par exemple, dans un cas où un code peut corriger des erreurs de poids maximal t, il est possible d'énumérer toutes les erreurs de poids au plus t $\{\vec{e_i}\}$ et de calculer leurs syndromes $\{\vec{s_i}\} = H\{\vec{e_i}\}$, dans ce cas il est possible d'obtenir des syndromes distincts pour chaque erreur distincte. Ensuite, pour la réception d'un message erroné $\vec{r}$, il suffit de calculer le syndrome associé à $\vec{r}$ et de localiser dans la table de syndrome l'erreur $\vec{e}$ correspondante. Finalement, il suffit de calculer $\vec{r} \oplus \vec{e}$ pour retrouver la chaîne de bits initiale. Cet algorithme est optimal dans le cas des codes linéaires. Par contre, la table de syndrome grandit de manière exponentielle avec la redondance $n - k$.

2.1.2 Un exemple : Le code de Hamming

Le code de Hamming [7,4,3], introduit par Richard Hamming en 1950 [22], s'agit historiquement du premier exemple non trivial de code de correction d'erreurs. Ce code permet d'encoder 4 bits de message $d_1d_2d_3d_4$ dans un mot de code de 7 bits. Pour ce faire, 3 bits de parités $p_1p_2p_3$ sont ajoutés en guise de redondance à la chaîne du message de 4 bits. Il en résulte une chaîne de 7 bits à transmettre $d_1d_2d_3d_4p_1p_2p_3$. La valeur des bits de parité est déterminée par des équations de contraintes qui sont illustrées simplement par le diagramme de la figure 2.1. Il faut que la somme des éléments de chacun des trois cercles soit paire. De cette manière, les 3 équations suivantes permettent de trouver p_1, p_2 et p_3,

$$p_1 \oplus d_1 \oplus d_2 \oplus d_3 = 0, \tag{2.3}$$
$$p_2 \oplus d_1 \oplus d_2 \oplus d_4 = 0, \tag{2.4}$$
$$p_3 \oplus d_1 \oplus d_3 \oplus d_4 = 0. \tag{2.5}$$

Ainsi, chacun des 2^4 mots de code possibles doit satisfaire ces équations. De plus, notons que chacune des erreurs possibles sur un des bits donne lieu à un ensemble de parités distinct pour chaque cercle, ce jeu de parité agit donc comme le syndrome de l'erreur. La table 2.1 présente les différentes valeurs de parités pour chaque cercle selon les erreurs de poids 1. Comme chacune des colonnes de cette table est distincte, toutes les erreurs de poids 1 peuvent être détectées et corrigées. En associant les vecteurs dans le code par $\vec{y} = (d_1\ d_2\ d_3\ p_1\ d_4\ p_2\ p_3)$, alors la table 2.1 se traduit directement comme étant la matrice de contrôle H pour le code de Hamming à 7 bits

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

La matrice de contrôle est utilisée pour le décodage, mais comment effectuer l'encodage ? Pour ce faire, il suffit de spécifier la matrice G. Il faut trouver la matrice

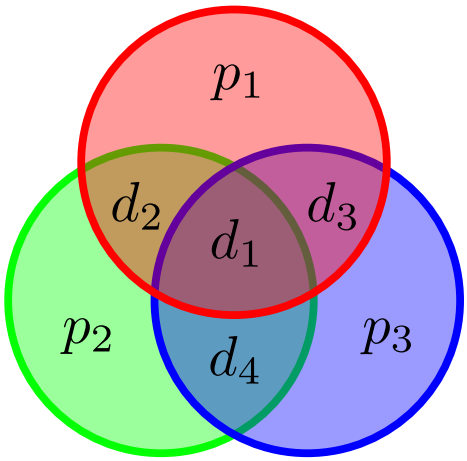


FIGURE 2.1 Diagramme représentant les équations de contraintes pour le code de Hamming à 7 bits. Il faut que la somme des bits dans chaque cercle soit paire.

Erreur	d_1	d_2	d_3	p_1	d_4	p_2	p_3
Rouge	1	1	1	1	0	0	0
Vert	1	1	0	0	1	1	0
Bleu	1	0	1	0	1	0	1

TABLE 2.1 Table de syndromes correspondant à la parité des 3 cercles pour les erreurs de poids 1.

G telle que $HG^T = 0$. Comme il s'agit d'un code linéaire, il est suffisant de calculer les valeurs de p_1, p_2 et p_3 pour les 4 vecteurs de base $(d_1\ d_2\ d_3\ d_4) = \{0001, 0010, 0100, 1000\}$ avec l'aide des équations ci-hautes. Ainsi, pour chaque vecteur de base, un vecteur $\vec{g} = (d_1\ d_2\ d_3\ p_1\ d_4\ p_2\ p_3)$ correspondant à une ligne de la matrice d'encodage G est obtenu. En effectuant ce calcul, la matrice G est donnée par,

$$G = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

Avec cette description pour G et pour H, la condition $HG^T = 0$ est vérifiée. Quelle est la distance minimum du code ? Il s'agit du poids de Hamming minimum parmi tous les mots de code à l'exclusion du mot de code $\vec{0}$. Comme les mots de code sont des combinaisons linéaires des lignes de la matrice G. Ceci permet de trouver une distance de $d = 3$. Ainsi, le code de Hamming est un code [7, 4, 3], il permet donc de corriger toutes les erreurs de poids 1 et la détection des erreurs de poids au plus 2. Quel est le gain que procure le code de Hamming de 7 bits vis-à-vis le code à répétition à 3 bits ? En effet, il semble que le code de Hamming et le code à répétition R_3 ont exactement le même pouvoir de correction et de détection d'erreurs. Par contre, le code de Hamming procure un avantage au niveau du rendement. Le code à répétition à 3 bits procure un rendement $R_{rep} = \frac{1}{3}$, alors que le code de Hamming permet un rendement $R_H = \frac{4}{7}$. Comme $R_H > R_{rep}$, le code de Hamming nécessite moins d'usage du canal que le code à répétition pour transmettre une quantité d'information donnée. Pour plus de détails sur la théorie des codes, les références suivantes constituent de bonnes introductions : [24], [19], [5], [23].

2.2 Théorie de l'information

Pour bien comprendre la famille de codes considérée dans ce mémoire, il est important d'introduire quelques concepts reliés à la théorie de l'information. Cette théorie majoritairement développée par Claude Shannon en 1948 [36] concerne les limites quant aux taux de compression et de transmission de l'information. L'introduction de cette théorie de la communication est fortement liée au développement de l'ère digitale et par le fait même de toutes les technologies qui en découlent. Cette section débute en discutant d'une mesure de l'information contenue dans une variable aléatoire appelée l'entropie. Ensuite, les canaux bruyants nécessaires à la modélisation du bruit sont traités, puis le second théorème de Shannon, établissant le taux d'information optimal qu'il est possible de transmettre à travers un canal bruyant, est présenté.

2.2.1 L'entropie au sens de Shannon

Comment mesurer la quantité d'information contenue dans une variable aléatoire? La notion d'entropie définit par Shannon donne une réponse à cette question.

Définition 8 *Soit un ensemble d'évènements discrets* $X = \{x_1, x_2, ..., x_n\}$ *muni d'une distribution de probabilité* $P = \{p_1, p_2, ..., p_n\}$ [4]*, alors l'entropie de* X *est donnée par*

$$H(X) = -\sum_{i=1}^{n} p_i \log p_i \text{ [5]}. \tag{2.6}$$

Définition 9 *Soit* (X, Y) *deux variables aléatoires avec une distribution de probabilité*

4. Lorsque le contexte est clair, la notation suivante $p(x_i) = p_i$ est utilisée, de même pour les probabilités jointes $p(x_i, y_j) = p_{ij}$. De plus, les variables aléatoires sont représentées en majuscule, alors que les valeurs que peuvent prendre celles-ci sont représentées en minuscule.

5. Le logarithme est en base 2, l'entropie se mesure donc en bits.

jointe donnée par p_{xy}, alors l'entropie jointe de (X, Y) est donnée par,

$$H(X,Y) = -\sum_{x=1}^{n_x}\sum_{y=1}^{n_y} p_{xy}\log p_{xy}. \tag{2.7}$$

L'*entropie conditionnelle* est obtenue en moyennant sur $H(X|Y=y)$,

$$H(X|Y) = -\sum_{x,y}^{n_x,n_y} p_{xy}\log\frac{p_{xy}}{p_x}. \tag{2.8}$$

Une des manières d'interpréter l'entropie consiste à considérer cette quantité comme une mesure du degré de surprise ou d'incertitude que procure une variable aléatoire en la mesurant. Ainsi, suivant cette interprétation, l'entropie est nulle pour une variable entièrement déterministe puisqu'elle ne procure aucune incertitude sur sa valeur. Au contraire, si la variable est maximalement aléatoire, l'entropie est maximale. Soit une variable aléatoire X de cardinalité [6] $|X|$ avec une distribution de probabilité uniforme $P_x = \frac{1}{|X|}$, alors $H(X) = \log|X|$. Cette valeur est le maximum pour H. L'entropie d'une variable aléatoire est donc bornée par

$$0 \leq H(X) \leq \log|X|. \tag{2.9}$$

Dans le cas de l'entropie conditionnelle, l'incertitude sur une variable aléatoire X est toujours plus grande ou égale à l'incertitude sur cette même variable aléatoire conditionnée sur une autre variable Y,

$$0 \leq H(X|Y) \leq H(X). \tag{2.10}$$

Une autre identité intéressante est la *règle de la chaîne* qui relie l'entropie jointe à une somme contenant l'entropie conditionnelle et l'entropie d'une variable de la manière suivante,

$$H(X,Y) = H(X) + H(Y|X). \tag{2.11}$$

Par symétrie,

$$H(X,Y) = H(Y) + H(X|Y). \tag{2.12}$$

6. La cardinalité d'une variable aléatoire est la taille de l'ensemble des valeurs que peut prendre cette variable aléatoire.

De plus, en combinant la règle de la chaîne avec l'équation 2.10, l'inégalité suivante est obtenue

$$H(X,Y) \leq H(X) + H(Y). \tag{2.13}$$

En guise d'exemple, soit $X = Bern(p)$, une variable de Bernoulli, alors $P(X = 0) = 1 - p$ et $P(X = 1) = p$. Par définition de l'entropie,

$$H(X) = h(p) = -(1-p)\log(1-p) - p\log p \tag{2.14}$$

Cette quantité est appelée la fonction d'entropie binaire et elle est maximale pour $p = \frac{1}{2}$, le cas où la probabilité est identiquement distribuée. Aussi, elle est minimale pour $p = 0$ ou $p = 1$, correspondant aux cas déterministes.

Pour étudier la quantité d'information que contient une variable aléatoire par rapport à une autre, le concept d'*information mutuelle* entre deux variables aléatoires est utilisé.

Définition 10 *Soit X et Y deux variables aléatoires avec une densité de probabilité jointe p_{xy} et avec probabilités marginales p_x et p_y. Alors, l'information mutuelle est*

$$I(X:Y) = \sum_x \sum_y p_{xy} \log \frac{p_{xy}}{p_x p_y}. \tag{2.15}$$

Il s'agit en quelque sorte d'une mesure de la corrélation entre 2 variables aléatoires. Cette quantité se décrit aussi en termes d'entropie,

$$I(X:Y) = H(X) + H(Y) - H(X,Y). \tag{2.16}$$

Par ailleurs, l'information mutuelle est une quantité symétrique

$$I(X:Y) = I(Y:X). \tag{2.17}$$

Dans le cas où les deux variables aléatoires sont indépendantes au sens où $P(X|Y) = P(X)$ et $P(Y|X) = P(Y)$, noté $X \perp Y$, la mesure de la variable Y ne révèle aucune information sur X et vice-versa. L'information mutuelle est donc nulle $I(X:Y) = 0$. La figure 2.2 illustre un diagramme de Venn des différentes quantités concernant l'entropie et l'information mutuelle de 2 variables aléatoires X et Y.

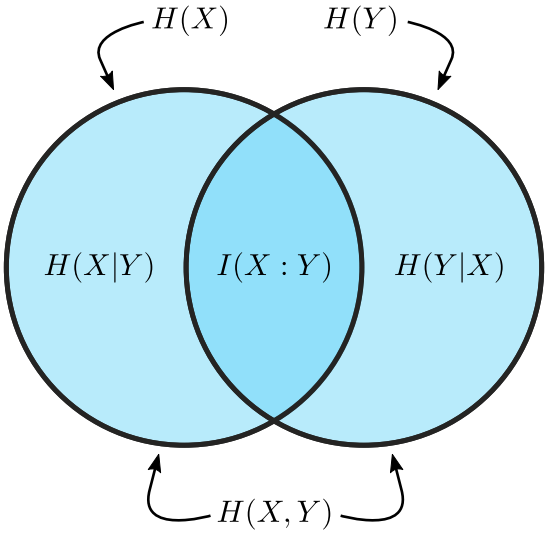


FIGURE 2.2 Diagramme de Venn illustrant les quantités H et I pour deux variables aléatoires X et Y.

2.2.2 Canal bruyant

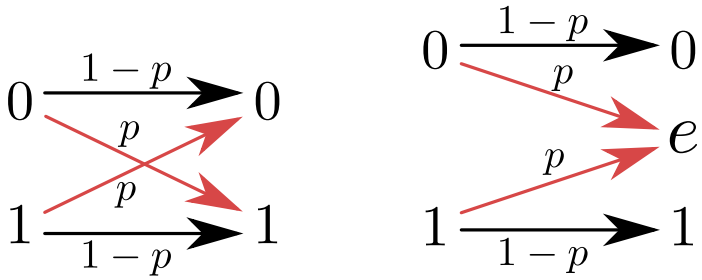
Afin de transmettre de l'information d'un point A à un point B (spatialement ou temporellement), il est nécessaire d'utiliser un support de transmission tel que l'air ou un fil de cuivre par exemple. Cet environnement peut parfois corrompre l'information en ajoutant du bruit. En théorie de la communication, une façon formelle de décrire ces environnements est d'utiliser la notion de canal bruyant.

Définition 11 *Un canal binaire est défini comme un système comportant une entrée binaire $X \in \{0, 1\}$, une sortie à valeur discrète Y et une matrice de probabilités de transitions $W(Y|X)$.*

Dans ce mémoire, le *canal binaire symétrique* constitue un exemple important, il s'agit d'un canal où $Y \in \{0, 1\}$ avec la matrice de transition suivante,

$$W(Y|X) = \begin{pmatrix} p & 1-p \\ 1-p & p \end{pmatrix}. \tag{2.18}$$

Le paramètre p représente la probabilité d'un renversement de bit. La notation $CBS(p)$ est utilisée pour définir un canal binaire symétrique avec probabilité d'erreur



(a) $BSC(p)$ **(b)** $BEC(p)$

FIGURE 2.3 Exemples de canaux binaires symétriques.

p. Il est possible de représenter graphiquement ce canal par un diagramme illustrant l'action du canal avec les probabilités de transitions respectives. La figure 2.3 a) présente le cas du canal binaire symétrique. Un autre canal d'importance est le *canal à effacement*. Comme son nom l'indique, ce canal prend en entrée un bit d'information et donne à la sortie un des trois états $\{0, 1, e\}$ où e indique que le bit d'entrée est effacé. Ceci survient avec une probabilité d'effacement p. La matrice de transition pour ce canal est donc,

$$W(Y|X) = \begin{pmatrix} 1-p & 0 & p \\ 0 & 1-p & p \end{pmatrix}. \tag{2.19}$$

La notation $CBE(p)$ est utilisée pour le canal à effacement avec probabilité d'erreur p. La figure 2.3 b) présente le diagramme du canal à effacement. Ces deux canaux comportent deux caractéristiques communes, ils sont *symétriques* et *sans mémoire*. Ils sont symétriques, car envoyer le bit 0 ou 1 produit la même probabilité d'erreurs. De plus, ils sont sans mémoire, car la probabilité jointe d'obtenir la chaîne de bits $\vec{y}$ sachant que la chaîne $\vec{x}$ est envoyée peut s'écrire comme

$$W_N(\vec{y}|\vec{x}) = \prod_{i=1}^{N} W(y_i|x_i). \tag{2.20}$$

Les résultats du reste de cette section concernent le cas des canaux sans mémoire. Il sera question des canaux avec mémoire à la prochaine section.

(a) Canal parfait **(b)** Canal poubelle

FIGURE 2.4 En a), l'information mutuelle est maximale et en b) elle est minimale.

2.2.3 Second théorème de Shannon

Les notions d'entropie et d'information mutuelle présentées à la sous-section 2.2.1 s'avèrent être des outils très utiles pour l'étude des canaux. Grâce à ces outils, la question suivante peut être posée : *quelle est la quantité maximale d'information qu'il est possible de transmettre à travers un canal bruyant ?* La réponse à cette question est au coeur du second théorème de Shannon, un des théorèmes les plus importants en théorie de l'information. En guise d'exemple, soit deux canaux limites, le canal parfait et le canal poubelle présentés à la figure 2.4. L'idée est d'utiliser la théorie de l'information pour quantifier la qualité d'un canal en fonction des paramètres d'entrée X et de sortie Y. L'information mutuelle $I(X : Y)$ s'avère être un candidat idéal pour mesurer la qualité d'un canal. Dans le cas du canal parfait, une mesure de l'information à la sortie donne la totalité de l'information sur l'entrée donc $I(X : Y)$ est maximale et donne donc dans ce cas 1 bit. Pour le canal poubelle, il est impossible d'obtenir de l'information sur les bits en entrée sachant la sortie, car il s'agit d'un effacement à tout coup donc l'information mutuelle est minimale, c'est-à-dire $I(X : Y) = 0$. Qu'en est-il pour les canaux $CBS(p)$ et $CBE(p)$? La *capacité d'un canal* est défini comme la quantité maximale d'information qu'il est possible d'obtenir sur la distribution d'entrée sachant celle de la sortie. Plus formellement,

Définition 12 *La capacité d'un canal est donnée par* $C = \max_{p(x)} I(X : Y)$.

Cette définition permet de calculer la capacité du canal binaire symétrique et du canal à effacement. Il est possible de montrer que $C_{CBS}(p) = 1 - h(p)$ où $h(p)$ est l'entropie binaire. Dans le cas du canal à effacement, la capacité est $C_{CBE}(p) = 1 - p$.

Ainsi, pour faire le lien avec les 2 cas limites, le canal parfait est simplement un $CBS(p = 0)$. Suivant la formule de la capacité, ce canal a donc une capacité de 1 bit. Alors que le canal poubelle est donné par $CBE(p = 1)$ ayant une capacité de 0 bit. La notion de capacité d'un canal peut être reliée à celle du rendement d'un code de correction d'erreurs, il s'agit du second théorème de Shannon.

Théorème 2 *(Second théorème de Shannon) Soit W un canal bruyant ayant une capacité C, alors il existe un code de correction d'erreurs tel que si le rendement est $R < C$, alors la probabilité d'erreurs diminue exponentiellement avec n la taille du code.*

Le second théorème de Shannon apporte donc une solution au problème soulevé en introduction quant à la recherche de *bons codes de correction d'erreurs*. Les codes qui peuvent satisfaire le théorème de Shannon avec un rendement R aussi près voulu de la capacité C du canal considéré sont appelés les *codes qui atteignent la capacité*. Il faut noter que la recherche d'un tel code est très difficile. En fait, à l'époque de Shannon (1948), aucun code de la sorte n'était connu. La preuve du théorème de Shannon n'est pas constructive, c'est-à-dire qu'elle ne donne pas de recette explicite pour la construction d'un code atteignant la capacité. Depuis 1948, les théoriciens des codes ont tenté sans succès de démontrer que certains codes pouvaient atteindre la capacité. Ce n'est qu'en 2009 avec l'introduction des *codes polaires* qu'on a pu avoir une famille de codes avec une preuve d'atteinte de la capacité et une complexité d'encodage et de décodage réaliste pour des applications pratiques [3]. Il sera question de cette famille de code dans quelques sections. Cette introduction à la théorie de l'information est majoritairement basée sur l'article initial de Shannon [36] et sur le livre de Cover et Thomas [11].

2.3 Canal avec mémoire et technique de codage

Cette section traite d'un type de bruit avec mémoire, c'est-à-dire un bruit qui n'agit plus indépendamment sur chacun des bits. Dans ce cas,

$$W_N(\vec{y}|\vec{x}) \neq \prod_{i=1}^{N} W(y_i|x_i). \tag{2.21}$$

Ce type de bruit est très intéressant puisqu'il est représentatif de plusieurs situations qui peuvent survenir lors de la transmission d'informations. Par exemple, la communication entre un satellite et la Terre où la puissance du signal envoyé est affectée par l'effet de masque (shadowing) dû aux obstacles physiques [34], ou bien simplement une rayure sur un disque compact. Il existe plusieurs types de canaux avec mémoire, la famille de canaux nommée les *canaux avec mémoire à états finis* sont considérés dans ce mémoire.

2.3.1 Canal avec mémoire à états finis

Un canal à d états avec mémoire est défini par un ensemble de d canaux binaires $W_s(y|x)$ avec $s \in \{1, 2, ...d\}$ sur lesquels il existe une dynamique de transition entre les états caractérisée par un processus stochastique. En spécifiant un temps i, un état s_{i-1} au temps antérieur et un bit d'entrée x_i, alors le bit à la sortie y_i et le prochain état s_i sont obtenus avec probabilité $W(y_i, s_i|x_i, s_{i-1})$. Le cas où le bit de sortie y_n est indépendant du prochain état s_n conditionnellement à l'entrée x_n et à l'état précédent s_{n-1} est considéré, c'est-à-dire que $y_n|x_n, s_{n-1} \perp s_n|x_n, s_{n-1}$. Ceci permet donc d'écrire

$$W(y_n, s_n|x_n, s_{n-1}) = p(y_n|x_n, s_{n-1})q(s_n|x_n, s_{n-1}). \tag{2.22}$$

Ces types de canaux peuvent donc être décrits par deux processus. Le premier décrit ce qu'il se passe au niveau de l'information par la densité de probabilité p et le second décrit les transitions d'états par la densité de probabilité q. De manière plus précise, la transition dans les états est décrite par une chaîne de Markov [7]. Le terme

7. Voir l'annexe A pour une introduction sur les chaînes de Markov.

de transitions d'états dépend du symbole transmis x_n. Dans ce cas, ce type de canal est appelé un *canal d'interférence intersymbole*. Il s'agit d'un modèle où le symbole transmis précédemment x_{n-1} affecte le symbole reçu au temps présent y_n. C'est un modèle de bruit bien connu en télécommunications. Lorsque $q(s_n|x_n s_{n-1}) = q(s_n|s_{n-1})$, les probabilités de transitions des états du modèle sont indépendantes de l'entrée. Ce type de canal est appelé un *canal de Markov à états finis*. Ces types de canaux vont être considérés pour la suite puisqu'ils constituent de bons modèles pour les bruits en rafales et peuvent être décrits simplement en utilisant les réseaux de tenseurs.

De manière générale, la probabilité jointe $W_N(\vec{y}|\vec{x})$ est une quantité non définie pour un canal à états finis sur N bits. Il faut spécifier un état initial s_0 pour le canal. Ainsi, la quantité $W_N(\vec{y}, s_N|\vec{x}, s_0)$ où s_N représente l'état final du canal est plutôt utilisée. Cette probabilité s'obtient récursivement

$$W_N(\vec{y}, s_N|\vec{x}, s_0) = \sum_{s_{N-1}} W(y_N, s_N|x_N, s_{N-1}) W_{N-1}(\vec{y}, s_{N-1}|\vec{x}, s_0). \tag{2.23}$$

En utilisant cette récurrence,

$$W_N(\vec{y}, s_N|\vec{x}, s_0) = \sum_{s_1^{N-1}} \prod_{n=1}^{N} W(y_n, s_n|x_n, s_{n-1}). \tag{2.24}$$

Finalement, en calculant la probabilité marginale sur s_N

$$W_N(\vec{y}|\vec{x}, s_0) = \sum_{s_1^{N}} \prod_{n=1}^{N} W(y_n, s_n|x_n, s_{n-1}). \tag{2.25}$$

Cette formule est générale et s'applique à tout type de canaux avec mémoire à états finis. Dans le cas d'un canal à interférence intersymbole cette formule peut s'écrire

$$W_N(\vec{y}|\vec{x}, s_0) = \sum_{s_1^{N}} \prod_{n=1}^{N} p(y_n|x_n, s_{n-1}) q(s_n|x_n, s_{n-1}). \tag{2.26}$$

Pour un canal de Markov à états finis la densité de probabilité est

$$W_N(\vec{y}|\vec{x}, s_0) = \sum_{s_1^N} \prod_{n=1}^{N} p(y_n|x_n, s_{n-1}) q(s_n|s_{n-1}). \tag{2.27}$$

Pour plus de détails sur ces canaux [19] et [21] constituent de bonnes références.

2.3.2 Capacité du canal avec mémoire à états fini

Le calcul de la capacité d'un canal avec mémoire est en général très difficile. Toutefois, pour le cas du canal avec mémoire à états finis, sous certaines conditions il est possible de simplifier les calculs et obtenir une bonne approximation de la capacité. Pour ce faire, il faut définir la capacité minimale et la capacité maximale d'un tel canal.

Définition 13 *La capacité minimale $\underline{C}$ est définie comme*

$$\underline{C} = \lim_{N\to\infty} \frac{1}{N} \max_{p(\vec{x})} \min_{s_0} I(\vec{X} : \vec{Y}|s_0) = \lim_{N\to\infty} \underline{C_N}. \tag{2.28}$$

Définition 14 *La capacité maximale $\overline{C}$ est définie comme*

$$\overline{C} = \lim_{N\to\infty} \frac{1}{N} \max_{p(\vec{x})} \max_{s_0} I(\vec{X} : \vec{Y}|s_0) = \lim_{N\to\infty} \overline{C_N}. \tag{2.29}$$

De manière générale, $\underline{C_N} \leq \overline{C_N}\ \forall\ N$. Dans le cas où $\underline{C} \neq \overline{C}$, il n'y a pas de notion claire de la capacité pour un tel canal. Pour obtenir $\underline{C} = \overline{C}$, il faut un canal où l'influence de l'état initial s_0 diminue avec le temps. Ce principe est la base de l'indécomposabilité d'un canal.

Définition 15 *Soit $q_N(s_N|\vec{x}, s_0) = \sum_{\vec{y}} W_N(\vec{y}, s_N|\vec{x}, s_0)$, alors ce canal est indécomposable si $\forall\ \epsilon > 0$, $\exists\ N_0$ tel que pour $N \geq N_0$*

$$|q_N(s_N|\vec{x}, s_0) - q_N(s_N|\vec{x}, s_0')| \leq \epsilon$$

$\forall\ s_N,\ \vec{x},\ s_0$ et s_0'.

Le théorème suivant donne une formule pour la capacité d'un canal indécomposable.

Théorème 3 *Soit un canal avec mémoire à états finis indécomposable, alors* $\underline{C} = \overline{C} = C$ *et la capacité de ce canal est donnée par*

$$C = \lim_{N\to\infty} \frac{1}{N} \max_{p(\vec{x})} I(\vec{X} : \vec{Y}). \tag{2.30}$$

En pratique cette quantité reste difficile à calculer puisqu'il existe en principe une quantité infinie et indénombrable de densité de probabilité sur la chaîne d'entrée $p(\vec{x})$ [4]. Dans le contexte des canaux de Markov à états finis, une condition nécessaire à l'indécomposabilité provient de l'ergodicité de la chaîne de Markov.

Théorème 4 *Soit un canal de Markov à états finis ayant une dynamique des états dont la chaîne de Markov est ergodique, alors ce canal est indécomposable.*

La notion d'indécomposabilité d'un canal, permet de généraliser le Second théorème de Shannon au cas des canaux avec mémoire à états finis. Ainsi, il est possible de transmettre de l'information sur un canal à états finis indécomposable avec une probabilité d'erreurs diminuant exponentiellement avec la taille du code n tant que le rendement R soit inférieur à la capacité C du canal [7].

2.3.3 Modèle de Gilbert-Elliott

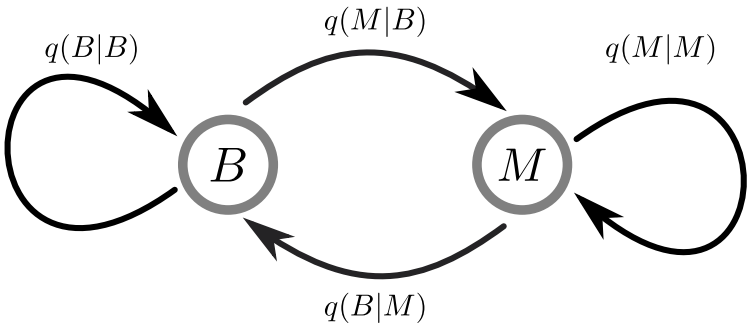


Figure 2.5 Graphe de transition du modèle de Gilbert-Elliott

Le modèle de Gilbert-Elliott est un des premiers modèles d'intérêts pour le bruit en rafale basé sur un canal de Markov à états finis [20], [14]. C'est sur ce modèle que les principaux résultats numériques de ce mémoire proviennent. Ce canal est basé sur une chaine de Markov à 2 états ayant des probabilités de transitions constantes. La figure 2.5 présente le graphe de transition du processus de Markov. L'espace des états est $\mathcal{S} = \{B, M\}$ avec B l'état *Bon* et M l'état *Mauvais*. Typiquement, les canaux binaires associés à l'état B et l'état M, notés respectivement W_B et W_M, sont des canaux binaires symétriques avec probabilité d'erreur h_B et h_M. La condition $h_B < h_M$ assure que l'état mauvais introduit plus d'erreurs que l'état bon. Par inspection de la figure 2.5, il est possible de voir que la chaîne de Markov est ergodique. Il existe donc un unique état stationnaire

$$\vec{\nu}_s = \frac{1}{q(M|B) + q(B|M)} \begin{pmatrix} q(B|M) \\ q(M|B) \end{pmatrix}. \tag{2.31}$$

La probabilité d'erreur moyenne du modèle de Gilbert-Elliott est donnée par

$$P_e = \frac{h_B q(B|M) + h_M q(M|B)}{q(M|B) + q(B|M)}. \tag{2.32}$$

Pour la simulation du bruit en rafale, il faut imposer une persistance dans les états B et M. Ceci est réalisé en fixant les probabilités de transitions $q(M|B)$ et $q(B|M)$ à des valeurs assez faibles. De plus, la longueur moyenne d'une chaîne de bit qui reste consécutivement dans un état donné, noté $\langle l_B \rangle$ pour l'état bon et $\langle l_M \rangle$ pour l'état mauvais est de distribution géométrique,

$$\langle l_B \rangle = \frac{1}{q(M|B)}, \quad \langle l_M \rangle = \frac{1}{q(B|M)}. \tag{2.33}$$

Il est possible de montrer que ce canal est indécomposable par la proposition suivante.

Proposition 1 *Soit un canal de Gilbert-Elliott, alors* $\forall\ \xi \in \{B, M\}$

$$q_l(s_l = \xi | s_0 = \xi) - q_l(s_l = \xi | s_0 \neq \xi) = (1 - q(B|M) - q(M|B))^l.$$

Cette proposition se prouve par induction sur l. Ce résultat montre que pour $|1 -$

$q(B|M) - q(M|B)| < 1$ le canal de Gilbert-Elliott est un canal indécomposable grâce à la définition 15.

Il est possible d'obtenir une formule simple pour la capacité du modèle de Gilbert-Elliott. Comme le modèle de bruit est indépendant de l'entrée $\vec{x}$ et de la sortie $\vec{y}$. Une chaine de bits $\vec{z}$ peut être générer de manière à décrire le bruit pour ensuite l'ajouter aux bits d'entrée $\vec{x}$ du canal. De cette manière, la chaîne de bits de la sortie du canal est obtenue par $\vec{y} = \vec{x} \oplus \vec{z}$ où l'opération $\oplus$ est appliquée bit à bit. Le processus stochastique décrivant la chaîne de bits $\vec{z}$ correspond à une chaîne de Markov cachée puisqu'elle ne donne aucune information sur les états. La variable aléatoire $Q_l = p(z_l = 1|Z_1^{l-1})$ [8] décrit la probabilité d'avoir une erreur au temps l sachant la chaîne d'erreur au temps antérieur. Cette quantité permet une expression pour le calcul de la capacité du canal de Gilbert-Elliott,

Proposition 2 *La capacité du canal de Gilbert-Elliott est donnée par*

$$C = 1 - \lim_{l \to \infty} \mathbb{E}[h(Q_l)] \tag{2.34}$$

avec $h(.)$ la fonction d'entropie binaire et $\mathbb{E}[.]$ l'espérance.

Il s'agit d'une formule pour la capacité beaucoup plus simple que la formule 2.30 étant donné qu'elle ne requiert plus de maximiser sur les densités de probabilité pour $\vec{X}$. Ce sujet est traité aux références suivantes [27] et [33].

2.3.4 Technique d'entrelacement

La technique d'entrelacement est une technique de codage couramment utilisée pour traiter les bruits avec mémoire [42], [41]. L'essence derrière cette technique est de transformer une chaîne de bits corrélée en une chaîne de bits non corrélée grâce à une permutation adéquate communément appelée un entrelaceur. L'utilisation d'un entrelaceur s'effectue normalement avec un code de correction d'erreurs qui offre de bonne performance pour du bruit sans mémoire. Un algorithme simple d'entrelacement est la technique d'entrelacement par bloc à L niveau. Il s'agit de

8. Z_1^{l-1} représente le vecteur $\vec{Z}$ dont la composante Z_l a été enlevée.

regrouper en une matrice L mots de code de manière à obtenir un mot de code par ligne. Une matrice $L \times n$ où n est la taille du code est ainsi obtenue. Ensuite, chaque colonne de cette matrice est transmise une à la suite de l'autre sur le canal avec mémoire. À la suite de la réception des mots de code, un décodage pour chaque ligne de cette matrice est effectué. Cette technique permet l'utilisation d'un code de correction d'erreurs standard pour les bruits avec mémoire. Soit un bruit possédant une rafale [9] de longueur l_r, alors une erreur d'au plus $\lceil \frac{l_r}{L} \rceil$ peut être produite sur chaque mot de code. Ainsi, un code de correction d'erreurs pouvant corriger les erreurs de poids $\lceil \frac{l_r}{L} \rceil$ pourra donc tolérer des erreurs avec rafale de longueur l_r. Pour illustrer le principe d'entrelacement, soit le code à répétition à 3 lettres défini pour l'alphabet avec le mot : *code* à envoyer. Il faut donc envoyer 4 mots de code : {*ccc*, *ooo*, *ddd*, *eee*}. Supposons un bruit en rafale de longueur $l_r = 4$ qui agit sur les 4 premiers symboles en changeant ces symboles par le symbole *x* et agit comme l'identité sur le reste des symboles. L'utilisation normale du code à répétition (a) est comparée avec l'utilisation d'un entrelaceur par bloc de niveau $L = 4$ (b).

$$a)\ ccc \mid ooo \mid ddd \mid eee \longrightarrow xxx \mid xoo \mid ddd \mid eee \longrightarrow xode$$

$$b)\ \begin{pmatrix} c & c & c \\ o & o & o \\ d & d & d \\ e & e & e \end{pmatrix} \longrightarrow \begin{pmatrix} x & c & c \\ x & o & o \\ x & d & d \\ x & e & e \end{pmatrix} \longrightarrow \begin{pmatrix} c \\ o \\ d \\ e \end{pmatrix}$$

En a), l'usage du code à répétition standard cause une erreur après décodage, car le mot $xode$ est décodé au lieu de $code$. Tandis qu'en b), l'usage d'un algorithme d'entrelacement permet de retrouver le message envoyé. En pratique, cette technique souffre d'un problème de latence relié au fait qu'il faut attendre d'avoir reçu les L mots de codes avant de pouvoir procéder au décodage. De plus, comme cette méthode ne tient pas en compte des caractéristiques internes du canal avec mémoire, il faut souvent s'attendre à observer une performance du décodeur sous-optimale.

9. Une rafale est une portion contiguë de la chaîne de bits qui contient les erreurs.

2.4 Les codes polaires

Les *codes polaires* sont une famille de codes de correction d'erreurs introduits en 2009 par E. Arikan [3]. Ceux-ci feront partie de la prochaine génération des standards de télécommunication (5G) [6]. Il existe une quantité énorme de littérature sur le sujet reliant diverses facettes des codes polaires dans le domaine des technologies des télécommunications. Dans cette section, il sera question d'une description plutôt sommaire des codes polaires en parlant du phénomène de polarisation, du circuit d'encodage et du décodeur. Les codes polaires font partie des classes de codes dits *modernes*, tel que les codes *LDPC* et les *turbo codes*. Contrairement au code de Hamming présenté à la section 2.1.2, où l'analyse du décodeur s'effectue facilement en étudiant les mots de code et la distance de Hamming, dans le cas des codes polaires, c'est tout autrement. En effet, les méthodes de décodage sont itératives et ne cherchent pas nécessairement à optimiser la distance de Hamming entre les différents mots de codes. Bien qu'il reste beaucoup de travail à faire pour caractériser complètement ces codes, ils sont les premiers codes de correction d'erreurs pouvant être démontrés mathématiquement à atteindre la capacité sur tous les types de canaux binaires symétriques [10] et ce avec une complexité d'encodage et de décodage relativement faible. C'est majoritairement cet aspect qui motive donc l'introduction des codes polaires dans les technologies du 5G malgré sa découverte récente.

2.4.1 L'idée de base

L'idée fondamentale des codes polaires repose sur une transformation simple. La figure 2.6 présente cette transformation sous forme de circuit binaire, il s'agit en fait de la porte *ou exclusif* accompagnée d'une copie du premier registre. Cette porte est nommée le *non contrôlé*. Soit 2 variables aléatoires binaires U_1 et U_2 en entrée du circuit. L'application de la transformation *non contrôlé* sur les variables aléatoires permet d'obtenir 2 autres variables aléatoires binaires X_1 et X_2. Ces deux nouvelles variables aléatoires sont transmises sur un canal bruyant composé de deux canaux W identiques pour les deux registres donnant les variables aléatoires Y_1 et Y_2. Les

10. Ils peuvent aussi atteindre la capacité pour les canaux de Markov à états finis. Ce cas est traité à la prochaine section.

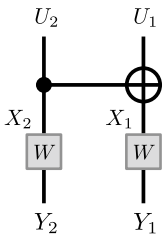


FIGURE 2.6 Transformation de base représentant le circuit du code polaire à 2 bits. L'encodeur prend les bits d'entrée U_1 et U_2 pour produire les bits X_1 et X_2. Ensuite, ces bits sont transmis sur 2 copies du canal W produisant les bits Y_1 et Y_2.

X_2X_1	U_2U_1
0 0	0 0
0 1	0 1
1 0	1 1
1 1	1 0

TABLE 2.2 Table de vérité pour la porte *non contrôlé* de la figure 2.6.

variables en U et en X peuvent donc être reliées par la transformation suivant la table de vérité 2.2. En termes d'équations,

$$U_1 = X_1 \oplus X_2, \quad U_2 = X_2 \tag{2.35}$$

Cette transformation est linéaire et inversible. Il est possible de quantifier l'effet de cette transformation en termes de l'entropie conditionnelle de la distribution d'entrée sachant la distribution de sortie correspondant à la quantité $H(X_1, X_2|Y_1, Y_2)$. Pour ce faire, deux canaux identiques W de type BEC ou BSC sont considérés. Sans la transformation, les deux couples de variables aléatoires (X_1, Y_1) et (X_2, Y_2) sont indépendants. Par ailleurs, comme il s'agit de canaux identiques, il faut que $H(X_1|Y_1) = H(X_2|Y_2)$ et donc $H(X_1, X_2|Y_1, Y_2) = 2H(X_1|Y_1)$. Comme la transformation 2.35 est bijective, elle ne doit pas changer l'entropie du schéma considéré, il s'agit d'une manifestation du principe de conservation de l'information d'un système. Alors, en considérant la transformation il faut que $H(X_1, X_2|Y_1, Y_2) = H(U_1, U_2|Y_1, Y_2)$. En utilisant la règle de la chaîne,

$$H(U_1, U_2|Y_1, Y_2) = H(U_1|Y_1, Y_2) + H(U_2|Y_1, Y_2, U_1). \tag{2.36}$$

Par ces considérations, le schéma de la figure 2.6 donne la relation suivante,

$$2H(X_1|Y_1) = H(U_1|Y_1, Y_2) + H(U_2|Y_1, Y_2, U_1). \tag{2.37}$$

En utilisant l'équation 2.10, $H(U_2|Y_1, Y_2, U_1) \leq H(U_2|Y_2)$. Comme $U_2 = X_2$,

$$H(U_2|Y_1, Y_2, U_1) \leq H(X_2|Y_2) = H(X_1|Y_1). \tag{2.38}$$

En remplaçant cette inégalité dans l'équation ci-haute,

$$2H(X_1|Y_1) \leq H(U_1|Y_1, Y_2) + H(X_1|Y_1) \rightarrow H(X_1|Y_1) \leq H(U_1|Y_1, Y_2). \tag{2.39}$$

En combinant 2.38 et 2.39

$$H(U_2|Y_1, Y_2, U_1) \leq H(X_1|Y_1) \leq H(U_1|Y_1, Y_2). \tag{2.40}$$

Cette inégalité est importante puisqu'elle montre l'effet de la transformation non contrôlé sur une paire de canaux identiques. En effet, avant l'ajout de la transformation, les canaux ont la même entropie conditionnelle. Par contre, après l'ajout de la transformation, une disparité dans les entropies conditionnelles est observée. L'idée fondamentale derrière les codes polaires consiste à tirer avantage de cette disparité pour l'encodage et le décodage de l'information. Cette procédure peut s'interpréter comme une transformation qui s'applique non pas sur les bits d'entrée, mais sur les canaux initiaux W, nommés canaux physiques, permettant d'en obtenir deux nouveaux W^+ et W^-, nommés canaux synthétiques. L'équation 2.40 permet de conclure que l'observation des quantités Y_1, Y_2, U_1 donne plus d'information sur $U_2 = X_2$ comparativement à ce que seul Y_2 peut donner. Cette situation peut donc être vue comme une méthode pour embellir ce canal. Ce canal est dénoté par W^+. Par contre, l'observation de Y_1, Y_2 donne moins d'information sur U_1 que ce que le canal initial W permet d'obtenir sur X_1 en observant Y_1. Ce canal est donc détérioré et il est noté par W^-. Cette disparité dans l'entropie conditionnelle des canaux provient du *phénomène de polarisation des canaux* dans les codes polaires. C'est donc ce phénomène qui permet de transformer 2 copies identiques d'un canal bruyant en un canal moins bruyant et un canal plus bruyant. Depuis l'invention des codes polaires, le phénomène de polarisation constitue un domaine de recherche comportant plusieurs applications en correction d'erreurs et en compression des données [2] et [44].

2.4.2 Le cas simple du canal à effacement

Le concept de polarisation des canaux s'illustre par un exemple concret. Pour ce faire, le schéma de la figure 2.6 est utilisé en supposant que $W = BEC(p)$. Soit deux canaux synthétiques $W^- : U_1 \to Y_1Y_2$ et $W^+ : U_2 \to Y_1Y_2U_1$ obtenus grâce à l'équation 2.40. Calculons la probabilité d'erreurs pour ces canaux. Comme les deux canaux physiques sont à effacement, il est facile de voir que W^- et W^+ sont aussi des canaux à effacement pour U_1 et U_2 respectivement. Ainsi, ayant les informations de sortie des canaux synthétiques, avec quelle probabilité est-il possible de retrouver le bit d'entrée ? Cette situation s'analyse en traitant le cas de W^- et W^+ séparément. Pour W^- : Il faut retrouver U_1 avec la connaissance de Y_1 et Y_2. Sachant que $U_1 =$

$X_1 \oplus X_2$ et que Y_1 et Y_2 sont affectés par les canaux physiques, la seule manière de retrouver U_1 est lorsque Y_1 et Y_2 ne sont pas effacés. Cet évènement survient avec probabilité $P_s = (1-p)(1-p)$. Ainsi, la probabilité d'effacement pour U_1 est de $P_e = 1 - P_s = 2p - p^2$. Ceci permet d'écrire, $W^- = BEC(2p - p^2)$. Pour W^+ : L'entrée est U_2 avec comme sortie Y_1, Y_2 et U_1. Comme $U_2 = X_2$, il est suffisant de connaître Y_2, mais grâce à la connaissance de U_1, ce n'est pas nécessaire. Ainsi, pour retrouver U_2 il faut considérer 3 possibilités. Soit, Y_1 et Y_2 ne sont pas effacés ce qui survient avec probabilité $P_1 = (1-p)(1-p)$. Soit, Y_1 est effacé et Y_2 n'est pas effacé ce qui survient avec probabilité $P_2 = p(1-p)$. Finalement, en utilisant la connaissance de $U_1 = X_1 \oplus X_2$ il est possible de retrouver U_2 même si Y_2 est effacé. Pour ce faire, il faut que Y_1 ne soit pas effacé, de cette manière X_1 est obtenu et donc comme $U_2 = X_2$, il est possible trouver $U_2 = Y_1 \oplus U_1$. Ceci survient avec probabilité $P_3 = p(1-p)$. Ainsi, la probabilité de succès est donc $P_s = P_1 + P_2 + P_3 = 1 - p^2$. La probabilité d'effacement pour ce canal est de $P_e = 1 - P_s = p^2$. Ceci permet d'écrire, $W^+ = BEC(p^2)$. Comme $p^2 \leq p \leq 2p - p^2$, alors W^- est un canal détérioré par rapport à W et W^+ est un canal embelli par rapport à W.

Une idée naturelle est de considérer cette procédure pour un plus grand nombre de canaux et d'ajouter des couches de transformations de manière à polariser les canaux les plus embellis ensemble. La figure 2.7 illustre un code polaire à 2 couches. Le nombre de couches est donné par $\log_2 N$ où N est le nombre de bits. Le cas à 1 couche se généralise donc pour une polarisation à 4 canaux. Cela permet d'obtenir les canaux synthétiques, $W^{--} : U_1 \to Y_1^4$, $W^{-+} : U_2 \to Y_1^4 U_1$, $W^{+-} : U_3 \to Y_1^4 U_1^2$ et $W^{++} : U_4 \to Y_1^4 U_1^3$. Les probabilités d'effacements pour ces canaux s'obtiennent de manière récursive à partir de ceux pour W^- et W^+, $W^{--} = BEC(2(2p - p^2) - (2p - p^2)^2)$, $W^{-+} = BEC(2p^2 - p^4)$, $W^{+-} = BEC((2p - p^2)^2)$ et $W^{++} = BEC(p^4)$. La figure 2.8 montre un exemple de polarisation pour le canal à effacement avec probabilité d'effacement de 0.5 pour les tailles $N = 32$ bits et $N = 256$ bits. Remarquons que les canaux synthétiques sont polarisés dans le sens où une certaine proportion tend vers une probabilité d'effacement de 0 et une autre proportion vers une probabilité d'effacement de 1. Aussi, ce phénomène est plus marqué pour un nombre de bits plus grand. Par contre, une certaine densité de ces canaux demeure dans une région où la probabilité d'effacement est $P_e \in (0, 1)$.

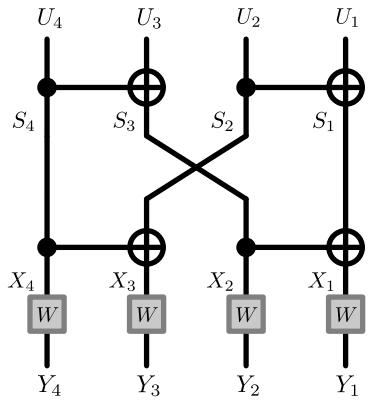
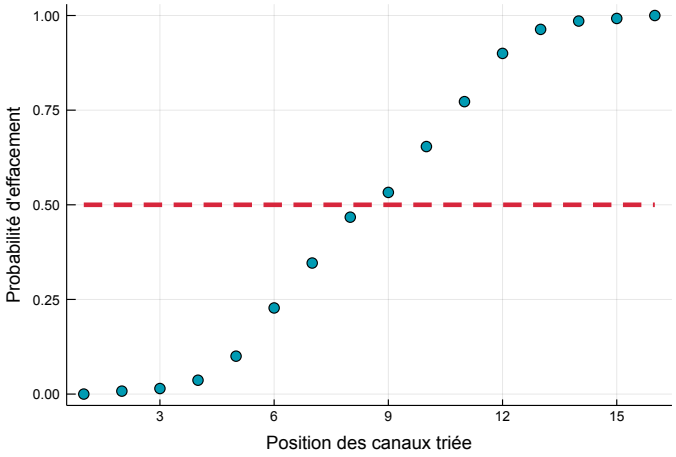
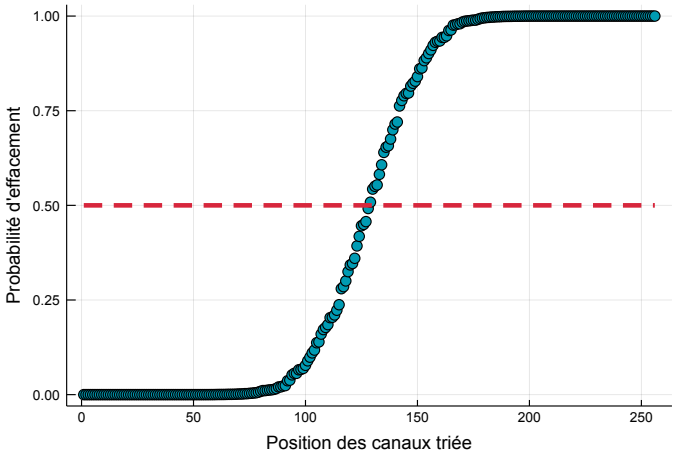


FIGURE 2.7 Construction d'un code polaire à 2 couches.



(a) Code polaire de 32 bits.



(b) Code polaire de 256 bits.

FIGURE 2.8 Polarisation du canal à effacement de probabilité d'effacement de 0.5.

2.4.3 Théorème de polarisation

De manière plus formelle, soit un code polaire à $N = 2^n$ bits où n est le nombre de couches avec N copies identiques et indépendantes d'un canal à entrée binaire W de capacité C. L'ensemble des chemins de polarisation associés aux canaux synthétiques est spécifié par $s^n \in \{+,-\}^n$. Ainsi, pour chaque couche, le canal est soit embelli $(+)$, soit détérioré $(-)$. Après n couches de polarisation, un canal synthétique est spécifié par W^{s^n}. Ces quantités sont utiles pour le théorème suivant,

Théorème 5 *(Polarisation) Pour tout canal à entrée binaire W de capacité C et pour tout $0 \leq a \leq b \leq 1$,*

$$\lim_{n\to\infty} \frac{1}{2^n} |s^n \in \{+,-\}^n : P(W^{s^n}) \in [0,a)| = C$$

$$\lim_{n\to\infty} \frac{1}{2^n} |s^n \in \{+,-\}^n : P(W^{s^n}) \in [a,b]| = 0$$

$$\lim_{n\to\infty} \frac{1}{2^n} |s^n \in \{+,-\}^n : P(W^{s^n}) \in (b,1]| = 1 - C$$

où $P(W^{s^n})$ représente la probabilité d'erreurs du canal synthétique considéré[11].

Ce théorème est surprenant pour 2 raisons. Premièrement, il assure que dans la limite d'un très grand nombre de canaux, la proportion de canaux avec des probabilités d'erreurs $\in (0,1)$ tend vers 0. Deuxièmement, la proportion de canaux synthétiques parfaits convergent vers la capacité du canal physique initial W. Ce théorème à lui seul est donc très important pour le codage utilisant les codes polaires puisqu'il permet de constater que les codes polaires atteignent la capacité. En effet, la technique de codage repose sur la classification des positions d'entrée $u_1, u_2, ...u_N$ pour lesquels les canaux synthétiques sont parfaits ou mauvais. L'idée consiste à envoyer de l'information seulement aux positions des canaux synthétiques parfaits. Les positions des canaux synthétiques mauvais servent donc à ajouter la redondance, typiquement ces bits d'entrée sont gelés à une valeur connue au décodeur. L'ensemble constitué des positions des bits gelés est noté $\mathcal{F}$, il s'agit d'un vecteur

11. Les canaux synthétiques dont la probabilité d'erreurs tend vers 0 sont appelés canaux synthétiques parfaits, alors que les canaux synthétiques ayant une probabilité d'erreurs qui tend vers 1 sont appelés canaux synthétiques mauvais.

contenant $N - k$ entiers distincts entre 1 et N. Ainsi, l'ensemble $\mathcal{F}^c$ contient les k positions contenant les bits d'information. Pour les bits gelés, la convention suivante est respectée : $u_i = 0 \; \forall \; i \in \mathcal{F}$.

2.4.4 Circuit d'encodage

Un circuit d'encodage pour les codes polaires de $N = 2^n$ bits comporte $n = \log_2 N$ couches. Chacune de ces couches comporte $\frac{N}{2}$ porte non contrôlé. Ainsi, au total le nombre de portes est $\frac{N \log_2 N}{2}$ pour un code de taille N. Le circuit d'encodage d'un code polaire à N bits est dénoté par $\mathcal{C}_N$. La construction du circuit d'encodage des codes polaires s'effectue récursivement de la manière suivante :

Cas de base : pour $\mathcal{C}_1$, il s'agit du circuit trivial [12].

Récursion : pour le circuit $\mathcal{C}_N$, la construction est basée sur une couche du bas, une permutation intermédiaire Π_N et une couche du haut. La couche du bas est composée de 2 copies de $\mathcal{C}_{\frac{N}{2}}$. Ensuite, une permutation Π_N particulière reliant la couche du bas et du haut est appliquée. Puis, la couche du haut est composée de portes non controlé appliquées à chaque paire de bits adjacents. La permutation Π_N envoie chaque bit en position impaire de la couche du haut vers la première moitié de la couche du bas, alors que chaque bit en position paire de la couche du haut est envoyé vers la seconde moitié de la couche du bas telle que le montre la figure 2.9.

Une fois que le circuit est construit, il ne reste plus qu'à sélectionner un ensemble de bits gelés. Pour atteindre un rendement de $R = \frac{k}{N}$, il faut sélectionner $N - k$ bits gelés de manière à pouvoir encoder k bits d'information. L'encodage est réalisé en propageant les bits en entrée du circuit à la sortie, ceci peut être fait en $O\left(\frac{N \log_2 N}{2}\right)$ opérations en comptant le nombre de portes d'un code polaire. En utilisant $\frac{N}{2}$ processeurs en parallèle, la complexité d'encodage en temps peut diminuer à $O(\log_2 N)$. Il est possible de définir une matrice d'encodage G pour les codes polaires, mais la représentation en circuit est beaucoup plus efficace. En effet, l'utilisation d'une matrice d'encodage demande habituellement une complexité d'encodage de $O(N^2)$ pour réaliser le produit matriciel nécessaire à l'encodage. Pour cette raison, la notation en circuit est préférée pour l'encodage.

12. Il s'agit du cas à 1 bit, ne comportant aucune porte logique.

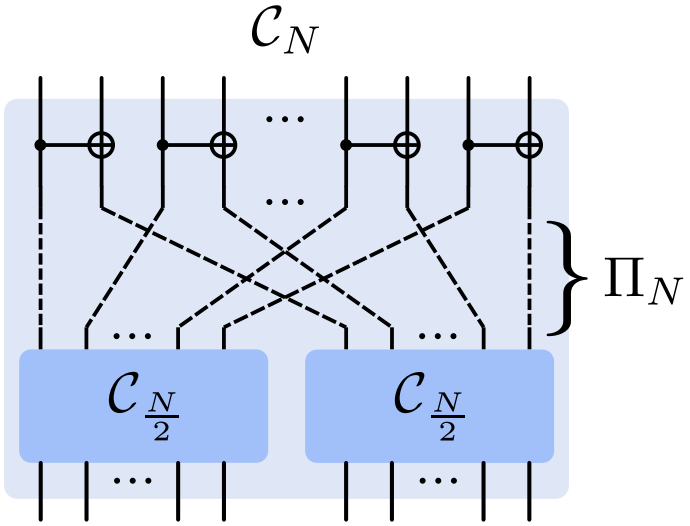


FIGURE 2.9 Construction récursive du circuit d'encodage $\mathcal{C}_N$ des codes polaires à partir de deux copies du circuit d'encodage $\mathcal{C}_{\frac{N}{2}}$. La couche du haut, composée de portes non controlé sur chaque bit adjacent, est connectée à la couche du bas via la permutation Π_N.

2.4.5 Décodeur par annulation successive

Soit un circuit d'encodage des codes polaires $\mathcal{C}_N$ obtenu en spécifiant N, k, un ensemble $\mathcal{F}$ de bits gelés et $\vec{u}$ l'entrée du circuit. Il est possible de générer le mot de code $\vec{x}$ en propageant $\vec{u}$ dans le circuit. Cette opération est dénotée par $\mathcal{C}_N(\vec{u}) = \vec{x}$. Ce mot de code est transmis dans un canal W_N qui est composé de N copies d'un canal binaire symétrique. À la sortie du canal, la chaîne $\vec{y}$ est observée. Le travail du décodeur est donc de trouver un estimé $\hat{\vec{u}}$ de $\vec{u}$ avec la connaissance de l'ensemble $\mathcal{F}$ et $\vec{y}$. Ainsi, un décodeur optimal consiste à calculer

$$\hat{\vec{u}} = \arg\max_{\vec{u}} W_N(\vec{y}\,|\mathcal{C}_N(\vec{u})). \tag{2.41}$$

Il s'agit de trouver l'entrée $\hat{\vec{u}}$ la plus probable parmi l'ensemble des 2^k entrées possibles. Un tel décodeur n'est toutefois pas réalisable en pratique pour des chaînes de bits de tailles considérables. Dans son article, Arikan propose un algorithme de décodage itératif appelé le *décodeur par annulation successive*. Ce décodeur décode un seul bit à la fois de droite à gauche du circuit en utilisant la connaissance des

bits déjà décodés, c'est-à-dire tous les bits à la droite de celui qui est à décoder et en négligeant complètement les bits à la gauche de ce bit, même s'ils sont préalablement dans l'ensemble $\mathcal{F}$. Ainsi, le décodage correspond à calculer pour chacune des N positions la fonction suivante :

$$\hat{u}_i = \begin{cases} 0 & \text{si } i \in \mathcal{F}, \\ \arg\max_{u_i} \sum_{u_{i+1}^N} W_N(\vec{y}\,|\mathcal{C}_N(\vec{u})) & \text{sinon.} \end{cases} \tag{2.42}$$

où la somme sur u_{i+1}^N représente une somme sur les composantes $i+1$ à N du vecteur $\vec{u}$ et les bits u_1 à u_{i-1} ont une valeur connue du décodeur puisqu'ils ont déjà été décodés. Ce décodeur est avantageux puisqu'il est possible de calculer $\sum_{u_{i+1}^N} W_N(\vec{y}\,|\mathcal{C}_N(\vec{u}))$ efficacement. Effectivement, Arikan a démontré des formules récursives permettant de calculer cette probabilité. Le décodage d'un code polaire de taille N requiert une complexité de calculs de $O(N \log N)$. Ce décodeur est sous-optimal dû notamment au fait qu'on ne permet pas la connaissance de tous les bits gelés lors du décodage. Malgré cela, Arikan a montré que son utilisation dans le contexte des codes polaires permet l'atteinte de la capacité asymptotiquement. Pour plus d'information sur les codes polaires [44] constitue une bonne référence.

2.4.6 Polarisation du bruit avec mémoire

Les codes polaires permettent-ils la polarisation du bruit avec mémoire ? Cette question, intéressante tant du point de vue pratique que théorique, n'est pas adressée dans l'article original des codes polaires, seulement les modèles de bruit indépendants sont traités. E. Şaşoğlu donne une réponse positive à cette question [43], [35] où il prouve en particulier que les canaux à états finis sont aussi polarisés par le circuit des codes polaires sans toutefois donner un décodeur pour cette situation. Ce résultat motive donc la recherche d'un décodeur efficace pour les codes polaires tenant en compte le bruit avec mémoire. C'est plus tard qu'il a été montré dans [40] un décodeur efficace pour les codes polaires soumis au bruit avec mémoire. Plus précisément, il s'agit d'une généralisation du décodeur à annulation successive de complexité de décodage de $O(|S|^3 N \log_2 N)$ avec S l'ensemble des états du canal, nommée décodeur par annulation successive sur treillis.

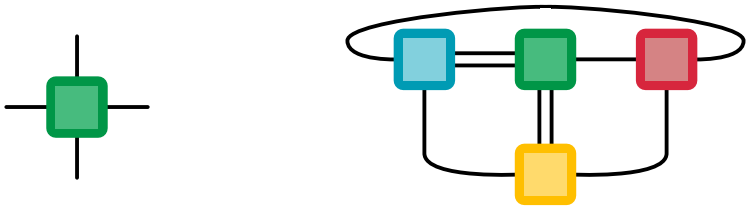
2.5 Les réseaux de tenseurs

En termes simples, les réseaux de tenseurs sont une structure de données particulière pour représenter certaines données corrélées de manière efficace. Par ailleurs, ces structures possèdent une interprétation graphique qui facilite leur usage semblable à ce que les diagrammes de Feynman procurent en physique de la matière condensée et en électrodynamique quantique. L'application principale des réseaux de tenseurs est en mécanique quantique, plus précisément dans l'étude des systèmes corrélés à N-corps telle une chaîne de spins. Ces outils permettent la mise en oeuvre d'algorithmes efficaces pour la simulation de ces systèmes sur un ordinateur classique [37]. Normalement, en mécanique quantique, pour représenter un état général $|\psi\rangle$ de N spins ou qubits, il faut un vecteur contenant 2^N amplitudes. Le problème de simulation d'un système quantique sur un ordinateur classique est donc intraitable puisqu'il requiert une mémoire classique grandissant de manière exponentielle avec la taille du système. Or il s'avère qu'en réalité, la plupart des systèmes quantiques d'intérêts n'ont pas besoin d'être spécifiés par un nombre exponentiel de paramètres [31], parfois un nombre polynomial de paramètres peut suffir auxquels cas, l'utilisation des réseaux de tenseurs pour représenter un tel système est bénéfique. Les réseaux de tenseurs sont aussi des outils utilisés pour la simulation de calculs quantiques [25] et l'étude des codes de correction d'erreurs quantiques [12, 13, 29, 9]. Récemment, les réseaux de tenseurs ont aussi été appliqués dans le domaine de l'apprentissage automatique [38]. Ils permettent notamment d'éviter le fléau de la dimension [13], phénomène bien connu en apprentissage automatique. De manière générale, les réseaux de tenseurs permettent une généralisation de l'algèbre linéaire. Dans le cadre de ce mémoire, il sera question d'utiliser les idées de bases des réseaux de tenseurs pour étudier le décodage d'un code de correction d'erreurs classique. Pour ce faire, il faut simplement comprendre comment manipuler ces objets. Pour plus d'information sur les réseaux de tenseurs appliqués en mécanique quantique [28] et [10] constituent de bonnes introductions.

13. Aussi connu sous le terme *curse of dimensionality*.

2.5.1 Définition

Un *tenseur* se définit comme un objet possédant des indices, avec le nombre d'indices définissant le *rang* du tenseur. De plus, à chacun de ces indices est associé une dimension interne. Selon cette définition, un vecteur est un objet défini par 1 indice, donc un tenseur de rang 1 alors qu'une matrice nécessite 2 indices [14] pouvant donc être décrit comme un tenseur de rang 2. De plus, un tenseur de rang 0 peut être compris comme un scalaire. Graphiquement, un tenseur de rang n se représente comme un noeud d'où émerge n arrêtes tel qu'illustré à la figure 2.10 a). De cette manière, un *réseau de tenseurs* est défini comme plusieurs noeuds connectés ensemble par des arrêtes. La figure 2.10 b) en présente un exemple. Ayant un tenseur ou un réseau de tenseurs, plusieurs opérations sont possibles. Notamment, le remodelage, la permutation des dimensions, la contraction, la décomposition en valeurs singulières, etc. Ces opérations sont utilisées de manière récurrente lorsqu'on traite des réseaux de tenseurs en physique du problème à N-corps.



(a) Un tenseur de rang 4.

(b) Réseau de tenseurs constitué de 5 tenseurs.

FIGURE 2.10

2.5.2 Remodelage

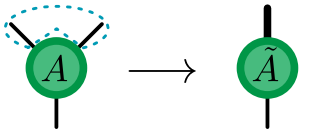
Le remodelage [15] d'un tenseur consiste en la modification de son rang. Un exemple simple illustrant bien le remodelage est la modification d'un vecteur (tenseur de rang 1) en une matrice (tenseur de rang 2). Il est facile de voir que pour un vecteur ayant une dimension interne première il est impossible d'effectuer un remodelage. Il existe deux types de remodelage, soit le *remodelage par groupement d'indices*

14. Un indice pour spécifier la ligne et l'autre pour la colonne
15. Dans la littérature, le remodelage fait référence au terme *reshape*.

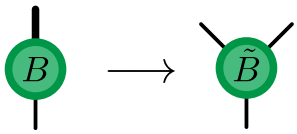
et le *remodelage par extension d'indices*. Le remodelage par groupement d'indices permet de réduire le rang du tenseur en augmentant du même coup la dimension interne d'un des indices. Par exemple, pour réduire le rang d'un tenseur de rang 3 A_{ijk} avec $\dim i = \chi_i$, $\dim j = \chi_j$ et $\dim k = \chi_k$ en combinant les deux indices i et j, il faut effectuer la transformation suivante $A_{ijk} \rightarrow A_{i(jk)} = A_{il}$. Ainsi, $\dim l = \chi_j \chi_k$, puis A_{il} est maintenant un tenseur de rang 2. Numériquement, il s'agit donc d'effectuer un produit d'indices donné par cette paramétrisation :

$$l = j + (k-1)\chi_j. \tag{2.43}$$

Il est possible d'effectuer l'opération inverse, c'est-à-dire augmenter le rang d'un tenseur en remodelant par extension d'indices. Par exemple, pour décrire un indice i de dimension χ_i avec 2 indices j et k de dimension respective χ_j et χ_k. Il faut que χ_j et χ_k soit des facteurs de χ_i. Ensuite, il suffit d'assigner les χ_j premiers éléments s en assignant les valeurs d'indices $j = s$ et $k = 1$. Par la suite, les éléments de $\chi_j + 1$ à $2\chi_j$ s'obtiennent de la même manière, mais en posant $k = 2$, et ainsi de suite jusqu'à la valeur d'indice $i = \chi_j \chi_k$ décrit par les indices $j = \chi_j$ et $k = \chi_k$. La figure 2.11 montre un exemple de ces deux types de remodelage.



(a) Remodelage par groupement d'indices d'un tenseur de rang 3.



(b) Remodelage par extension d'indices d'un tenseur de rang 2.

FIGURE 2.11 Illustration du principe de remodelage d'un tenseur.

2.5.3 Permutation des indices

La permutation des indices est une généralisation de la transposition d'une matrice appliquée pour un tenseur de rang $n \geq 2$. Pour un tenseur de rang n, une

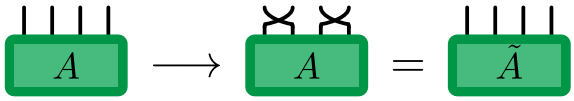


FIGURE 2.12 Exemple d'une permutation d'indices pour un tenseur de rang 4 A_{ijkl} soumis à la permutation $[2, 1, 4, 3]$ résultant en un nouveau tenseur $\tilde{A}_{ijkl}$.

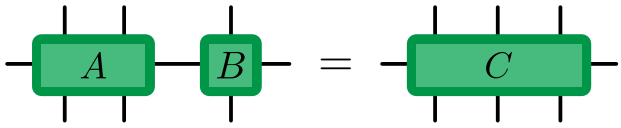


FIGURE 2.13 Illustration d'une contraction entre un tenseur A de rang 6 et un tenseur B de rang 4 résultant en un tenseur C de rang 8.

permutation des indices se définit par un vecteur de permutation de n éléments appliqué aux indices. Par exemple, pour un tenseur de 4 indices A_{ijkl}, la permutation $[2, 1, 4, 3]$ résulte en un tenseur A_{jilk}. La figure 2.12 illustre cet exemple.

2.5.4 Contraction d'un réseau de tenseurs

La contraction de tenseurs est une généralisation de la multiplication matricielle appliquée à des tenseurs d'ordre supérieurs. Par exemple, pour effectuer la multiplication d'une matrice A_{ij} avec une matrice B_{kl}, il faut avoir la condition $\dim j = \dim k$. Le produit matriciel s'écrit comme $\sum_j A_{ij} B_{jl}$. Ainsi, cette opération s'effectue en sommant sur l'indice j. La contraction de tenseurs est donc simplement obtenue en réalisant une somme sur les indices de tenseurs d'ordre supérieur. Graphiquement, la contraction entre un tenseur A de rang m et un tenseur B de rang n se représente comme un segment reliant une arrête du tenseur A à une arrête du tenseur B, pour autant que la dimension de ces arrêtes soit égales, il en résulte ainsi un tenseur C de rang $n+m-2$. La figure 2.13 montre un exemple de contraction. Cette représentation graphique de la contraction permet donc d'obtenir un réseau de tenseurs défini par plusieurs tenseurs interconnectés.

2.5.5 L'ordre de contraction

De manière générale, pour un réseau de tenseurs donné, il faut effectuer la contraction du réseau à la suite d'un algorithme d'optimisation ou de simplification résultant en la quantité cherchée. Pour définir un algorithme de contraction, il faut choisir un ordre de contraction, celui-ci peut avoir un impact considérable sur l'efficacité en ce qui concerne la mémoire et le temps d'exécution. Ceci est similaire à la réalisation d'un calcul comme $x(A + B)$. En considérant l'addition et la multiplication comme 1 unité de calcul, alors il est possible de calculer cette expression en distribuant la multiplication et effectuer l'addition, auxquels cas, 2 multiplications et 1 addition sont effectuées donc 3 unités de calcul. D'une autre part, il est plus efficace d'effectuer l'addition en premier et ensuite la multiplication ce qui donne 2 unités de calcul. Dans le cas d'un réseau de tenseurs, ce type d'optimisation est souvent crucial pour obtenir des résultats en temps et usage de mémoire décent. Il s'avère que le problème d'optimisation des étapes de contraction d'un réseau de tenseurs est très difficile en général [1]. En effet, ce problème est relié à la classe de complexité *NP-complet*. Il n'existe donc pas d'algorithme efficace qui prend en entrée un réseau de tenseurs et donne à la sortie les étapes de contractions optimales. Par contre, pour des instances particulières de réseaux de tenseurs, il est possible de résoudre ce problème facilement. Par exemple, les réseaux de tenseurs formant une chaîne ou un arbre peuvent être contractés de manière efficace. Voici un exemple qui montre l'importance de l'ordre de contraction. Pour contracter le réseau de tenseurs illustré à la figure 2.14, il existe plusieurs choix quant à l'ordre de contraction. Ces choix peuvent avoir un impact considérable concernant les coûts de calculs (temps) et le stockage (mémoire). Par exemple, l'ordre de contraction proposé à la figure 2.15 donne un temps de calcul dominé par $O(\chi^5)$, alors que pour l'usage de mémoire, il faut de l'ordre de $O(\chi^4)$ unité de mémoire. En comparant avec la séquence de contraction proposée à la figure 2.16, il faut un temps de calcul de l'ordre de $O(\chi^4)$ et une utilisation en mémoire de $O(\chi^3)$. Cet algorithme de contraction est donc plus efficace que le premier.

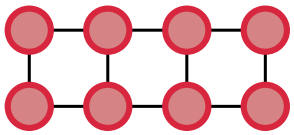


Figure 2.14 Réseau de tenseurs en forme d'échelle.

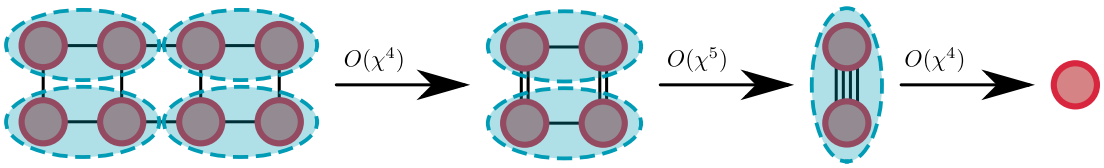


Figure 2.15 Contraction sous-optimale

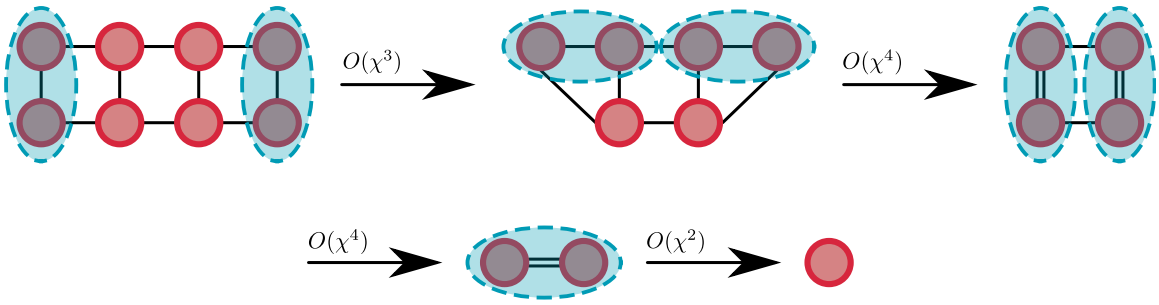


Figure 2.16 Contraction optimale

2.6 Les codes polaires convolutifs

Dans la section 3, il est décrit comment les réseaux de tenseurs peuvent être utilisés pour le décodage de certains codes de corrections d'erreurs. Cette découverte est due principalement à A. Ferris et D. Poulin où ils ont proposé ces outils pour le décodage de codes de corrections d'erreurs quantiques [17] applicables aussi en correction d'erreurs classique. Ils traitent aussi d'une généralisation possible des codes polaires inspirée par un réseau de tenseurs appelé le *branching MERA* introduit par G. Vidal et G. Evenbly [15]. Ainsi, grâce à cette connexion entre les réseaux de tenseurs et la correction d'erreurs, il a été possible de généraliser les codes polaires en les *codes polaires convolutifs*. Le circuit d'encodage des codes polaires est caractérisé par une structure interne par bloc. C'est-à-dire que pour une couche du circuit d'encodage des codes polaires, les portes non contrôlé sont appliquées par blocs indépendants comme le montre la figure 2.17. A. Ferris et D. Poulin ont montré qu'il est possible d'étendre les codes polaires en ajoutant une structure convolutive au circuit d'encodage tout en conservant les bonnes propriétés de décodage des codes polaires [18]. La structure convolutive est caractérisée par une couche dont les portes logiques ne sont plus appliquées par blocs indépendants. Cette situation est illustrée à la figure 2.18. Intuitivement, l'ajout de ces portes logiques permet un phénomène de polarisation plus marqué. En effet, pour le cas où $N = 4$, la structure du circuit des codes polaires convolutifs permet de combiner les canaux synthétiques W^{+-} et W^{-+} tels qu'illustrés à la figure 2.19, alors que ceci n'est pas le cas pour les codes polaires. Le cas à $N = 8$ bits est présenté à la figure 2.20, la construction du circuit reste récursive comme dans le cas des codes polaires. Ainsi, pour des tailles de codes plus élevées, une polarisation plus rapide des canaux que dans le cas de codes polaires standard devrait être observée. Effectivement, les comparaisons de performances entre les codes polaires et les codes polaires convolutifs effectuées dans [18, 39, 26] montrent que pour des modèles de bruit sans mémoire, les codes polaires convolutifs sont plus performants. Dans ce mémoire, ces résultats sont étendus aux cas de modèle de bruits avec mémoire [8].

FIGURE 2.17 Illustration d'une couche du circuit d'encodage des codes polaires. Chacune des portes logiques agit par bloc indépendant sur chaque paire de bits.

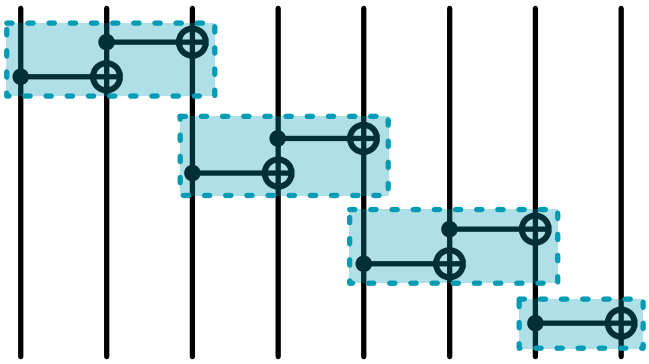


FIGURE 2.18 Illustration d'une couche du circuit d'encodage des codes polaires convolutifs. La structure n'est plus par bloc indépendant, mais plutôt par une structure convolutive.

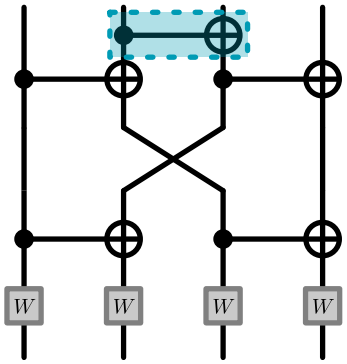


FIGURE 2.19 Code polaire convolutif pour $N = 4$ bits. La région ombragée bleue représente la différence entre le cas convolutif et le cas non convolutif.

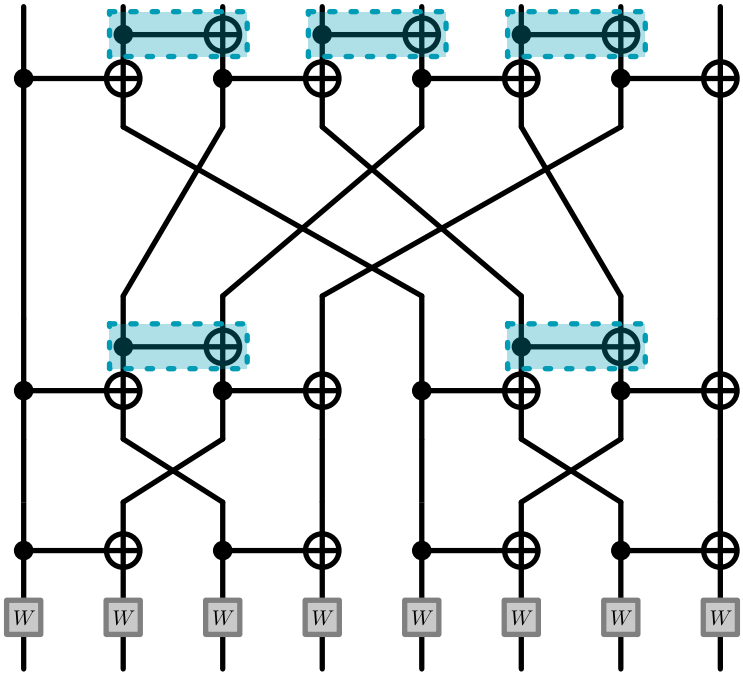


FIGURE 2.20 Code polaire convolutif pour $N = 8$ bits. La région ombragée bleue représente la différence entre le cas convolutif et le cas non convolutif.

Chapitre 3

Méthodologie

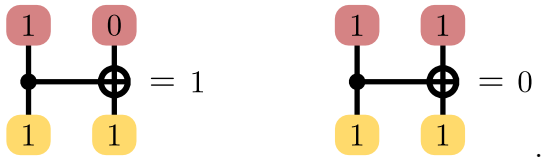
Dans ce chapitre, la méthodologie utilisée pour la réalisation du projet de recherche est introduite. Il s'agit d'appliquer les réseaux de tenseurs vus précédemment dans un contexte autre que celui de la mécanique quantique, soit en correction d'erreurs classique. Plus précisément, le but du projet de recherche est l'implémentation d'un décodeur efficace pour les codes polaires et codes polaires convolutifs soumis au bruit avec mémoire. Premièrement, une correspondance entre le problème du décodage et celui de la contraction d'un réseau de tenseurs est exposée. Ensuite, une formulation du décodeur par annulation successive en termes d'un algorithme faisant intervenir un réseau de tenseurs est donnée. Finalement, une description du modèle de bruit à états finis en termes d'un réseau de tenseurs est démontrée.

3.1 Formulation d'un circuit en termes des réseaux de tenseurs

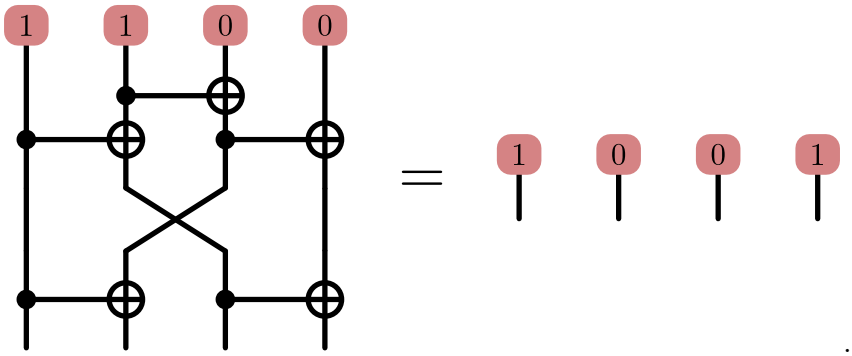
Un circuit d'encodage d'un code de correction d'erreurs peut se décrire comme un réseau de tenseurs. Pour ce faire, il suffit d'assigner les états de bits 0 et 1 à des tenseurs de rang 1

$$= \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{3.1}$$

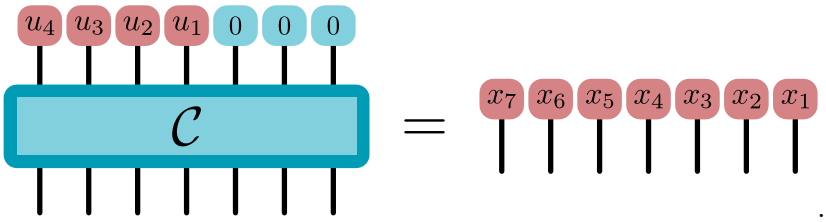
De cette manière, une porte logique représente un tenseur. Dans ce cas, la porte non contrôlé est un tenseur de rang 4 avec chacun des 4 indices représentant les bits d'entrée et de sortie. Ainsi, ce tenseur prend la valeur 1 lorsque les assignations d'indices correspondent aux assignations données par la table de vérité 2.2, autrement le tenseur vaut 0. Par exemple,



Ainsi, un ensemble de portes logiques représentant un circuit est décrit en termes d'un réseau de tenseurs. Par exemple, le circuit d'encodage d'un code polaire convolutif à $N = 4$ bits est représenté par un réseau de tenseurs contenant 5 tenseurs de rang 4. La contraction de ce réseau de tenseurs avec une chaîne de bits en entrée permet d'obtenir une chaîne à la sortie. Par exemple,



De manière générale, le circuit d'encodage est illustré comme un tenseur de rang $2N$ avec $N - k$ positions où les états de bits d'entrée sont gelés à la valeur 0, il s'agit de l'ajout de redondance au message. Par exemple,

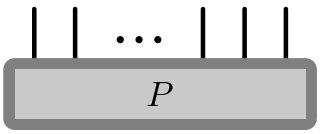


3.2 Le décodage, un problème de contraction

L'application des réseaux de tenseurs dans le contexte de la théorie des codes s'établit grâce à une correspondance entre le problème de contracter un réseau de tenseurs et le problème de décodage d'un code de correction d'erreurs. Cette correspondance permet l'implémentation du décodeur par annulation successive en termes de réseaux de tenseurs. Avant de développer cette correspondance, il est important de comprendre comment représenter et manipuler une densité de probabilité dans le langage des réseaux de tenseurs.

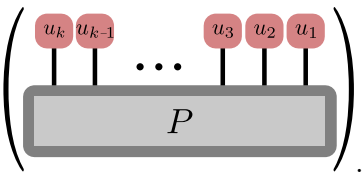
3.2.1 Calculs sur une densité de probabilité

Soit une densité de probabilité sur k bits $P(\vec{u})$, où $\vec{u}$ peut être interprété comme la chaîne de bits de message dans le contexte de la correction d'erreurs. Cet objet peut être décrit comme un vecteur contenant 2^k éléments. Celui-ci peut facilement s'écrire comme un tenseur de rang k avec chaque indice de dimension 2 représentant chaque bit obtenu en appliquant un remodelage par extension d'indices. Cette densité de probabilité peut donc se représenter comme

$P(\vec{u}) =$ 

. (3.2)

Pour trouver la chaîne de bits la plus probable $\vec{u}$, il faut utiliser le décodage par maximum de vraisemblance. Pour ce faire, il suffit de trouver l'argument qui maximise cette probabilité. Il s'agit donc de calculer

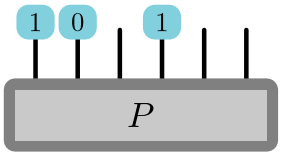
$\vec{\hat{u}} = \arg\max_{\vec{u}}$ 

. (3.3)

Ce problème d'optimisation est intraitable, c'est-à-dire qu'il faut un nombre de calculs exponentiel pour obtenir le résultat. En effet, il s'agit du même problème que

celui du décodage optimal de l'équation 2.41. Ainsi, pour des raisons d'efficacité, au lieu d'optimiser en fonction de la chaîne la plus probable, une technique d'optimisation itérative sur chacun des bits est utilisée. Cette technique est à la base du décodeur par annulation successive. Pour ce faire, il est important de montrer comment effectuer les calculs de probabilité conditionnelle et marginale sur la densité de probabilité. Pour calculer la probabilité conditionnelle $P(u_1, u_2, u_4 | u_3 = 1, u_5 = 0, u_6 = 1)$ à une constante de normalisation près, il faut effectuer la contraction,

$$P(u_1, u_2, u_3 = 1, u_4, u_5 = 0, u_6 = 1) = \quad . \tag{3.4}$$



Pour le calcul d'une probabilité marginale

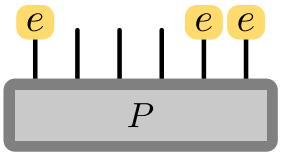
$$P(u_2, u_3, u_4) = \sum_{u_1, u_5, u_6} P(\vec{u}), \tag{3.5}$$

il s'agit de sommer sur les indices. Pour ce faire, le tenseur de rang 1 suivant est introduit,

$$= \begin{pmatrix} 1 \\ 1 \end{pmatrix} . \tag{3.6}$$

Cette probabilité marginale peut donc se calculer par l'équation suivante,

$$\sum_{u_1, u_2, u_6} P(\vec{u}) = \quad . \tag{3.7}$$

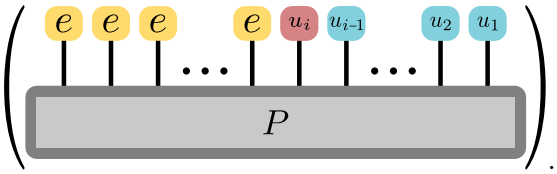


3.2.2 Décodeur par annulation successive

Il est donc possible de faire correspondre des calculs de probabilités marginales et conditionnelles à un problème de contraction d'un réseau de tenseurs. Le principe de base du décodeur par annulation successive s'illustre facilement en utilisant les réseaux de tenseurs. Tel qu'expliqué à la section 2.4.5, ce décodeur procède en décodant un seul bit à la fois de manière itérative en allant de droite à gauche.

Supposons une densité de probabilité $P(\vec{u})$ sur N bits. Pour maximiser chaque bit de droite à gauche du circuit, il faut calculer $\hat{u}_i = \arg\max_{u_i \in \{0,1\}} P(u_i|u_1, u_2, ..., u_{i-1})$ pour chacune des positions. Ainsi, en termes de réseaux de tenseurs cette équation s'écrit comme

$$\hat{u}_i = \arg\max_{u_i} \left(\right) .$$



(3.8)

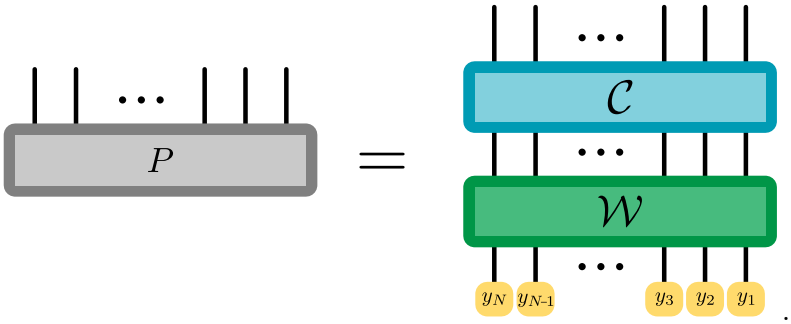
Cette manière de procéder est sous-optimale puisqu'elle n'optimise pas en fonction de la chaîne de bits globale, mais elle est beaucoup plus efficace puisqu'il n'y a que 2 valeurs de bits possibles. C'est sur ce principe qu'est basé le décodeur par annulation successive. Par contre, la contraction de ce tenseur n'est en général pas efficace. Dans le cas des codes polaires et des codes polaires convolutifs, il est possible de contracter ce réseau de tenseurs efficacement, les explications sont données à la section 3.4. Pour obtenir la chaîne totale, il faut donc effectuer N itérations de ce type. Jusqu'ici, l'obtention de la densité de probabilité sur les chaînes de bits a été supposée. Pour la suite, il est question de comment l'obtenir dans le contexte du décodage des codes polaires et des codes polaires convolutifs.

3.2.3 L'obtention de la densité de probabilité

Soit un circuit d'encodage $\mathcal{C}$ pour un code polaire ou un code polaire convolutif à $N = 2^n$ bits. Tel que vu précédemment, ce circuit contient des portes logiques et celles-ci peuvent être vues comme des tenseurs. Le circuit ainsi obtenu est donc un tenseur de rang $2N$ avec N indices pour les bits d'entrées du circuit et N indices pour les bits de sortie du circuit. Supposons un modèle de bruit binaire générique $\mathcal{W}$ agissant sur les N bits[1]. Ce modèle de bruit peut aussi se représenter par un tenseur de rang $2N$. Un remodelage de ce tenseur en une matrice 2^N par 2^N donne une matrice stochastique telle que la somme de chacune de ses colonnes donne 1. Puis, l'ensemble des bits reçus à la sortie du canal se représente par un ensemble de tenseurs de rang 1 $y_1, y_2, ..., y_N$. La densité de probabilité sur les chaînes de bits

1. Le modèle de bruit est traité plus en détail à la section suivante.

d'entrée possibles P est donc donnée par



(3.9)

Cette formulation est très générale et elle peut s'appliquer pour plusieurs circuits d'encodage $\mathcal{C}$ et modèle de bruit $\mathcal{W}$. Ainsi, la contraction de ce réseau de tenseurs conditionné sur les bits de redondance spécifiés au décodeur donne une densité de probabilité binaire sur les messages envoyés. Le décodeur à maximum de vraisemblance recherchera donc la chaîne de bit la plus probable. Bien entendu, ce cas générique est intraitable, mais il est possible de spécifier $\mathcal{C}$ et $\mathcal{W}$ pour lesquels ce schéma peut s'appliquer de manière efficace lorsqu'un décodeur par annulation successive est utilisé.

3.3 Modèle de réseaux de tenseurs pour canal bruyant

Dans cette section, il est présenté une formulation en termes de réseaux de tenseurs pour décrire les canaux de type binaire symétrique sans mémoire et les canaux avec mémoire à états finis.

3.3.1 Canaux sans mémoire

Tel que vu précédemment, un modèle de bruit binaire symétrique est décrit grâce à $W(Y|X)$, une matrice contenant les probabilités conditionnelles. Ainsi, formulé

en termes de réseaux de tenseurs, il suffit de représenter ce type de canal par un tenseur de rang 2 avec des indices représentant l'entrée X et la sortie Y

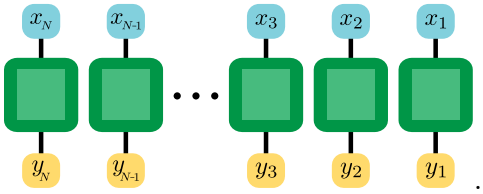
$$W(Y|X) = \quad . \tag{3.10}$$

De cette manière, les tenseurs représentant les états de bits peuvent être utiliser pour calculer la probabilité conditionnelle. Par exemple, pour un canal binaire symétrique de probabilité d'erreur p,

$$W(Y = 1|X = 0) = \quad = p. \tag{3.11}$$

Pour une séquence de N bits $\vec{x}$ soumise à N copies indépendantes de ce canal, la probabilité d'obtenir la chaîne $\vec{x}$ à la sortie du canal sachant l'entrée est donnée par $W(\vec{y}|\vec{x}) = \prod_{k=1}^{N} W(y_k|x_k)$, une caractéristique fondamentale d'un modèle de bruit sans mémoire. En termes de réseaux de tenseurs,

$$W(\vec{y}|\vec{x}) = \prod_{k=1}^{N} W(y_k|x_k) = \quad . \tag{3.12}$$



3.3.2 Canaux avec mémoire

Considérons le cas du canal avec mémoire à états finis avec d états en supposant que la dynamique répond à un modèle stochastique correspondant à une chaîne de Markov ergodique. Tel que vu à la section 2.3.1, un canal de Markov à états finis est

décrit par l'équation,

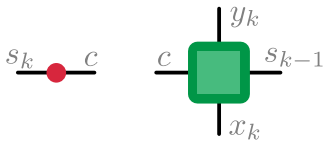
$$P_N(\vec{y}|\vec{x}, s_0) = \sum_{s_1^N} \prod_{k=1}^{N} q(s_k|s_{k-1}) p(y_k|x_k, s_{k-1}). \tag{3.13}$$

De manière équivalente,

$$P_N(\vec{y}|\vec{x}, s_0) = \sum_{s_1^N} \prod_{k=1}^{N} \sum_{c} q(s_k|c)\delta_{c,s_{k-1}} p(y_k|x_k, c). \tag{3.14}$$

Il est possible d'associer les tenseurs suivants à l'équation précédente,

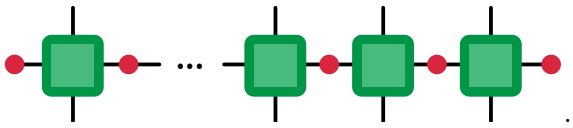
$$W_N(y_1^N|x_1^N, s_0) = \sum_{s_1^N} \prod_{k=1}^{N} \sum_{c} \underbrace{q(s_k|c)}\underbrace{\delta_{c,s_{k-1}} p(y_k|x_k, c)}$$



. (3.15)

Ainsi, en termes de réseaux de tenseurs un canal de Markov à états finis est décrit par le réseau de tenseurs suivant,

$$W_N(y_1^N|x_1^N, s_0) =$$



. (3.16)

Ce réseau de tenseurs possède une topologie en chaîne bien connue dans le domaine de la physique des systèmes quantiques à N-corps, il porte le nom d'opérateur à produit de matrices [2] [30]. De manière plus intuitive, ce type de canal est décrit par une chaîne combinant un tenseur de rang 2 décrivant la matrice de transition des états et un tenseur de rang 4 décrivant l'espace des canaux $W_C(Y|X)$. De plus, comme l'effet de ce bruit est étudié sur des chaînes de bits finies, deux tenseurs de rang 1 agissant comme conditions frontières sont définis. Le tenseur de rang 1 sur la frontière de droite représente la distribution initiale S_0 comme étant la distribution stationnaire décrite par le vecteur $\vec{\nu}_s$. Puis, le tenseur de rang 1 sur la frontière de

2. Dans la littérature, ce terme fait référence à *matrix product operator (MPO)*.

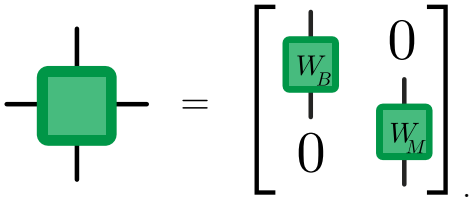
gauche représente une somme sur tous les états finaux S_N, encodant le fait qu'à la fin du processus l'état du canal C peut prendre n'importe quelle valeur. Cette somme est réalisée par un vecteur dont l'ensemble des composantes est 1. Ces deux conditions frontières sont représentées par,


$= \vec{\nu}_{s,}$ (3.17)


$= \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}.$ (3.18)

Dans le cas du modèle de Gilbert-Elliott, les tenseurs sont définis comme


$= \begin{bmatrix} q(B|B) & q(B|M) \\ q(M|B) & q(M|M) \end{bmatrix},$ (3.19)



. (3.20)

3.4 Algorithme de décodage

Les sous-sections précédentes montrent qu'il est possible d'utiliser les réseaux de tenseurs pour décrire le circuit d'encodage et le modèle de bruit. En principe, les réseaux de tenseurs peuvent être utilisés pour effectuer un décodage selon l'algorithme par annulation successive pour un code polaire ou un code polaire convolutif soumis au bruit avec mémoire. Dans cette sous-section, il est question d'appliquer ces outils de manière efficace. Pour le décodage d'un code de la famille des codes

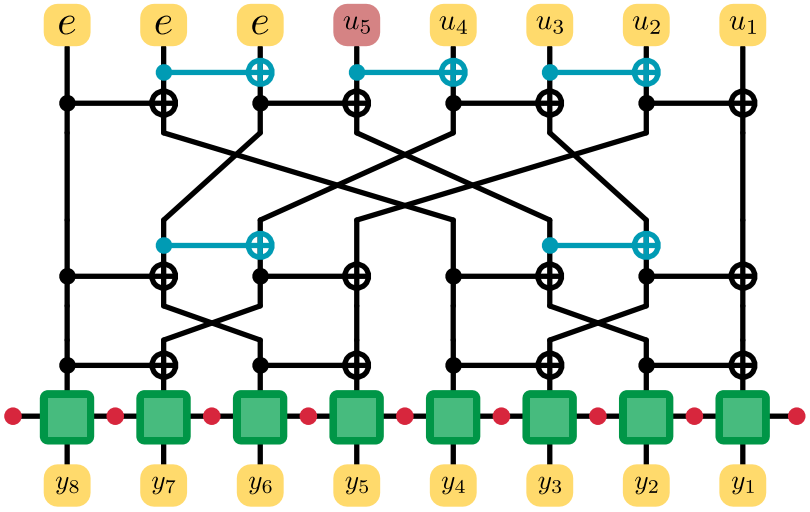


FIGURE 3.1 Réseau de tenseur illustrant une étape du décodeur par annulation successive. Les bits u_1 à u_4 sont supposés connus du décodeur. La contraction de ce tenseur donne un vecteur contenant les probabilités des valeurs pour le bit u_5.

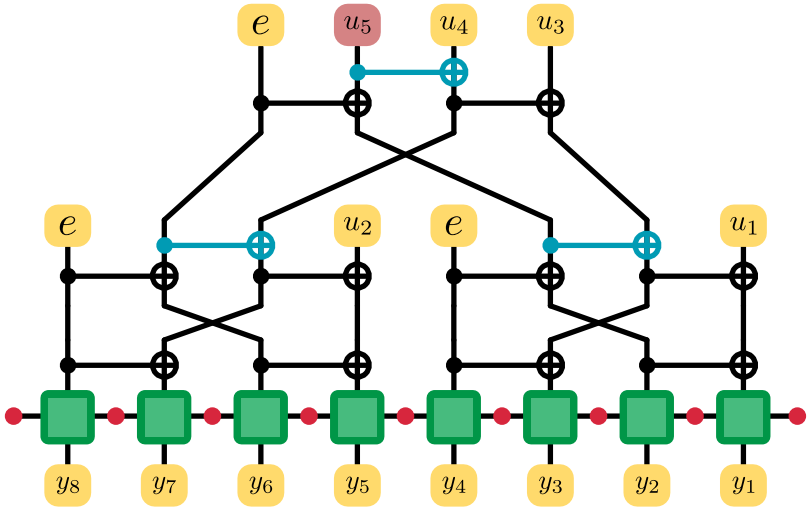


FIGURE 3.2 Illustration de l'étape de simplification du réseau de tenseurs utilisé pour le décodage du bit u_5.

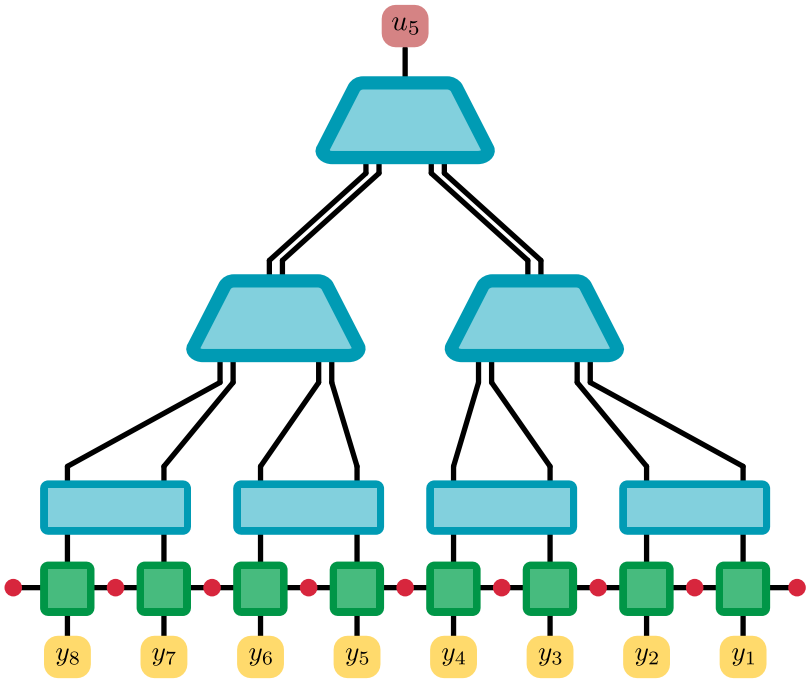
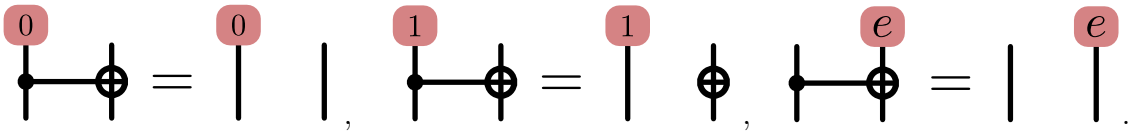


FIGURE 3.3 Après simplification, le réseau de tenseurs obtenu correspond à un circuit ayant la topologie d'un arbre dont les feuilles sont attachées horizontalement par une chaîne due au modèle de bruit avec mémoire.

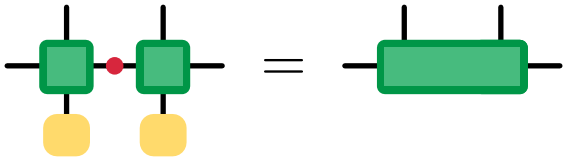
polaires [3], un ensemble de bits gelés $\mathcal{F}$ est supposé connu, ces bits sont fixés à la valeur 0 par convention. En utilisant la théorie développée à la section 4.2, on note qu'il suffit de spécifier un circuit $\mathcal{C}$ et un modèle de bruit $\mathcal{W}$. La figure 3.1 présente un exemple du décodage du bit u_5 par l'algorithme d'annulation successive pour un code polaire (portes logiques noires) et un code polaire convolutif (portes logiques noires et bleues) de taille $N = 8$ bits dans un contexte de bruit avec mémoire. A priori, la contraction du réseau de la figure 3.1 semble difficile, mais il est possible d'utiliser des identités algébriques simplifiant grandement le circuit. Il s'agit des simplifications suivantes,



(3.21)

3. Le terme famille des codes polaires fait référence aux généralisations possibles telles que celle du code polaire convolutif.

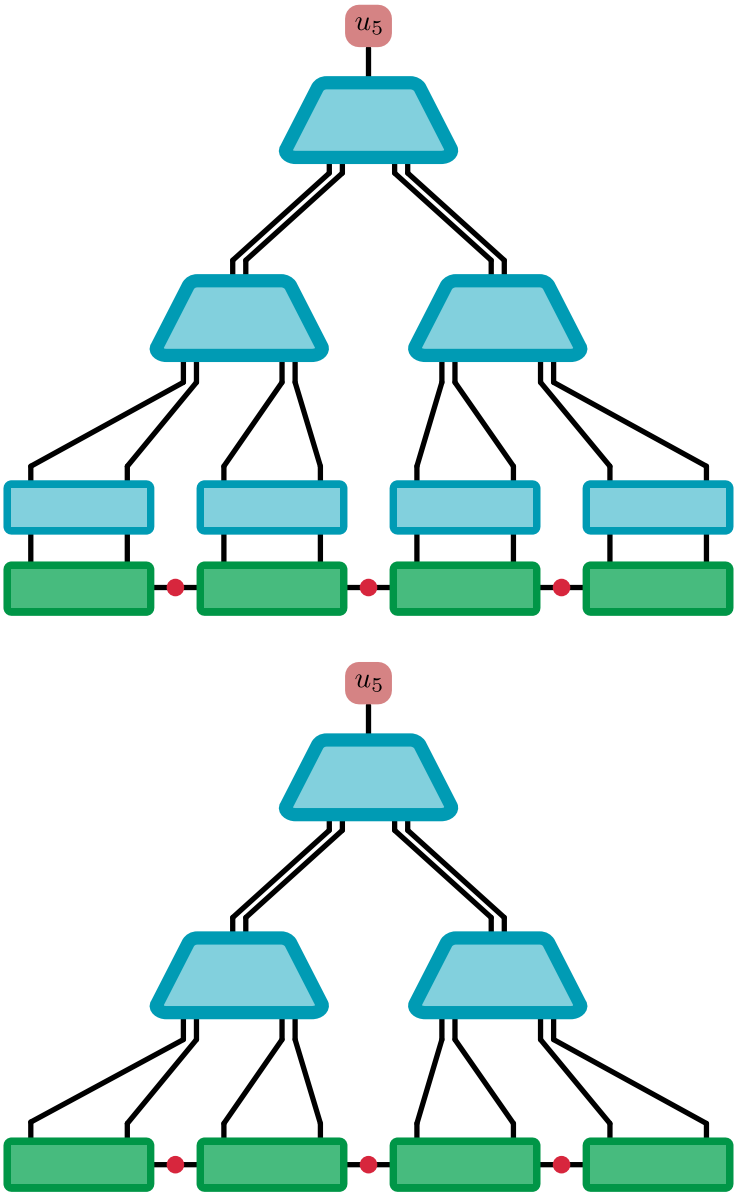
Ainsi, l'utilisation de ces simplifications couplées au décodeur par annulation successive permet une contraction efficace du réseau de tenseurs et donc un décodage efficace. Typiquement, l'algorithme de décodage pour un bit donné s'effectue en 2 étapes, soit une étape de simplification du circuit grâce à l'utilisation des identités, suivie d'une étape de contraction des tenseurs du bas vers le haut. La figure 3.2 montre un exemple de simplification du circuit pour $N = 8$ bits. Cette simplification permet d'obtenir un circuit ayant la même topologie qu'un réseau de tenseurs en arbre dont les feuilles sont connectées à un réseau de tenseurs ayant la topologie d'une chaîne correspondant au bruit avec mémoire comme le montre la figure 3.3. Une fois les simplifications complétées, il ne reste plus qu'à contracter ce réseau de tenseurs. Un réseau de tenseur ayant la topologie d'un arbre de profondeur $\log N$ peut se contracter efficacement en contractant du bas vers le haut résultant en une complexité de calculs de $O(N)$. Dans le cas du réseau de la figure 3.3, il ne s'agit pas complètement d'un arbre puisque les feuilles sont connectées horizontalement. Par contre, on peut adapter la contraction du bas vers le haut d'un arbre en effectuant une contraction intermédiaire entre chacune des paires de feuilles connectées au bas du circuit. Il s'agit spécifiquement d'effectuer cette contraction

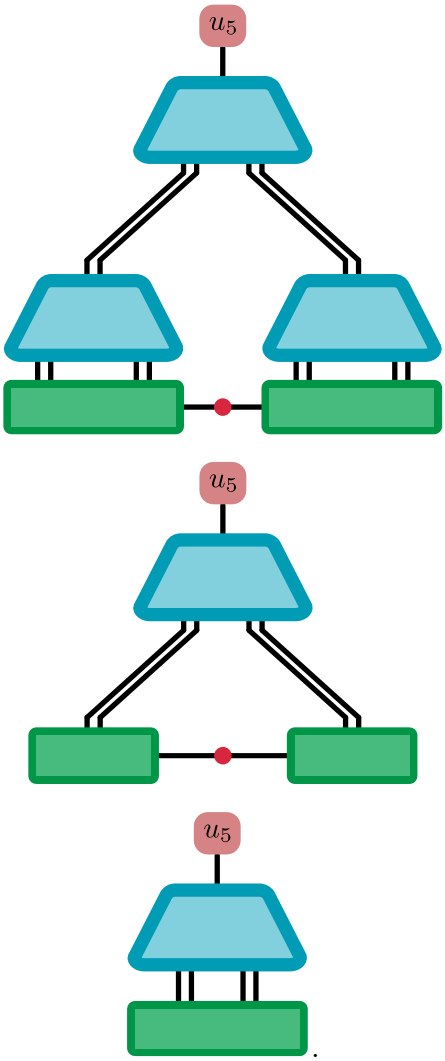


. (3.22)

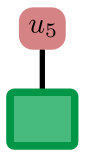
Comme la dimension de l'indice horizontal est d, représentant le nombre d'états du canal avec mémoire, on trouve que le coût de cette contraction est $O(d^3)$. Ainsi,

partant du circuit de la figure 3.3, la série de contraction suivante est effectuée :





Finalement, il suffit de calculer

$$\hat{u}_5 = \underset{u_5 \in \{0,1\}}{\arg\max} \left(\right.$$


$$\left. \right).$$

De manière générale, le coût de contraction pour un réseau de tenseurs ayant un circuit en arbre dont les feuilles sont attachées à un modèle de bruit avec mémoire ayant la topologie d'une chaîne est de $O(d^3 N)$. Dans l'exemple donné ici, il s'agit seulement du décodage d'un seul bit. Il faudrait donc itérer la procédure de décodage pour les N positions du circuit, en effectuant une procédure de simplification et de

contraction à chaque fois. Lors d'une implémentation du décodeur, il est possible de recycler plusieurs étapes de calculs reliées à la simplification et à la contraction du circuit de sorte qu'au total l'algorithme nécessite une complexité de calculs de $O(d^3 N \log N)$.

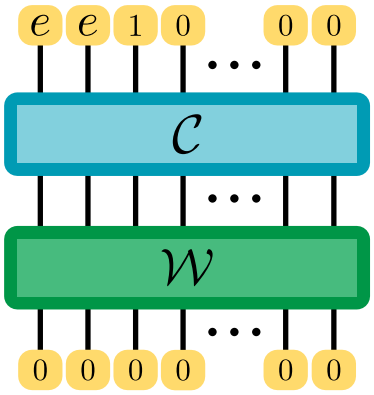
3.5 Sélection des bits gelés

Une des étapes cruciales pour la construction de bons codes dans la famille des codes polaires est la sélection des bits gelés. Cette sélection est déterminante pour obtenir un code atteignant la capacité. Intuitivement, la position des bits gelés devrait être aux endroits où les canaux synthétiques obtenus par le phénomène de polarisation sont le plus médiocres. Ainsi, cette sélection dépend du modèle de bruit. Dans l'article original d'Arikan, une sélection des bits gelés basée sur le paramètre de Battacharyya d'un canal $Z(W)$ est utilisée. Il s'agit d'une quantité qui mesure la distance entre deux densités de probabilité reliées à l'entrée et la sortie du canal. Il s'avère que pour le canal à effacement, cette quantité se calcul exactement. Par contre, pour les autres types de canaux, tel que le canal binaire symétrique, le calcul de $Z(W)$ est intraitable. Les réseaux de tenseurs offrent une méthode alternative pour déterminer la position des bits gelés. L'algorithme consiste à évaluer la probabilité $E(i)$ d'obtenir une erreur indétectable en position i sachant qu'aux positions 1 à $i-1$ aucune erreur n'a eu lieu. Cette probabilité est évaluée pour chacune des positions de 1 à N. Ensuite, ce vecteur de probabilité est classifié de manière à geler $N(1-R)$ bits aux positions qui ont les probabilités d'une erreur indétectable les plus élevées.

De par son interprétation en termes de réseaux de tenseurs, il est facile d'appliquer cet algorithme autant pour le cas du bruit sans mémoire qu'avec mémoire. Il faut seulement s'assurer que le réseau résultant est efficacement contractable. Pour ce faire, il suffit de poser l'ensemble des bits reçus à 0 puis on applique la même méthode que pour le décodage par annulation successive, donc un décodage de droite à gauche. Cette fois-ci par contre, on emmagasine en mémoire la probabilité p_1, correspondant à la probabilité d'une erreur non détectée, pour chaque bit décodé

et on pose le bit décodé à la valeur 0. L'équation suivante présente un exemple pour le calcul de $E(N-2)$

$$E(N-2) = \quad . \tag{3.23}$$



La complexité de cet algorithme est similaire à celle du décodage, soit de $O(N \log N)$. Ainsi, avant d'effectuer les simulations numériques pour l'évaluation de la performance d'un code on utilise cette sous-routine pour trouver $\mathcal{F}$.

Chapitre 4

Résultats et analyse

Ce chapitre donne les détails concernant les simulations effectuées dans le contexte du projet de maîtrise concernant l'étude du bruit avec mémoire dans les codes polaires et les codes polaires convolutifs. La performance du décodeur décrit dans la section précédente y est présentée.

4.1 Simulations

L'évaluation de la performance du décodeur s'effectue grâce à des simulations de type Monte-Carlo. Pour ce faire, une chaîne de bits est échantillonnée selon le modèle de bruit. Puis elle est décodée selon le type de décodeur. Ensuite, la chaîne de bits initiale est comparée avec la chaîne de bits décodée. Typiquement, deux mesures de performances peuvent être réalisées. Soit le taux d'erreurs par bit (BER *Bit Error Rate*) et le taux d'erreurs par trame (bloc) (FER *Frame Error Rate*). Dénotons par N_e le nombre de bits ayant une erreur après le décodage. Par construction du décodeur, il faut que $N_e \leq K$ car les $N - K$ autres bits sont fixés à la valeur 0. Le *BER* est défini comme le ratio $\frac{N_e}{K}$. Dans le cas du taux d'erreurs par trame, la chaîne complète après le décodage est considérée et si au moins une erreur survient sur ce bloc alors cette quantité prend la valeur 1. Si aucune erreur n'est survenue alors le *FER* est 0. Le *FER* constitue donc une borne supérieure sur le *BER*. En pratique, un nombre d'itérations N_{iter} est effectué ce qui permet de calculer le *BER* moyen et le

FER moyen. Intuitivement, le *FER* moyen correspond à la probabilité d'erreur après décodage. En comparaison, le taux d'erreur sans décodage est $P_e = 1 - (1 - P_{moy})^k$ en considérant la probabilité d'erreur moyenne du canal. De plus, compte tenu de l'atteinte de la capacité des codes polaires et des codes polaires convolutifs, on s'attend à observer une diminution de la probabilité d'erreur après décodage avec N. Idéalement, cette suppression devrait survenir de manière exponentielle, mais il faut rappeler que l'atteinte de la capacité demeure une preuve valide dans le domaine asymptotique où $N \to \infty$. Évidemment, les résultats numériques obtenus sont à échelle finie pour des tailles d'intérêts allant de 16 à 2048 bits. La prochaine section montre les résultats obtenus pour divers canaux avec mémoire dans le cas des codes polaires et des codes polaires convolutifs.

4.2 Résultats

On rappelle que le modèle de Gilbert-Elliott est une chaîne de Markov à 2 états B et M avec $W_B = BSC(h_B)$ et $W_M = BSC(h_M)$. Pour les simulations, on fixe $h_B = 0$ et $h_M = h$. Ainsi, ce modèle contient 3 paramètres libres, soit $q(B|M)$, $q(M|B)$ et h. Le ratio

$$\rho = \frac{P(B)}{P(M)} = \frac{q(B|M)}{q(M|B)}, \tag{4.1}$$

quantifie le temps moyen dans l'état *B* par rapport à l'état *M*. Dans la limite où $\rho \to \infty$ et $\rho \to h$, le modèle de bruit tend respectivement vers un canal sans bruit et un canal $BSC(h)$. Les simulations effectuées concernent 3 types de décodeurs. Le premier type de décodeur (*sans mémoire*) correspond à l'algorithme de décodage standard basé sur le décodeur par annulation successive introduit par Arikan ne tenant pas en compte la structure du bruit avec mémoire. Ce décodeur utilise seulement la probabilité d'erreur moyenne du bruit comme paramètre. Le second type de décodeur (*permutation*) utilise l'algorithme standard accompagné d'une technique d'entrelacement produisant des permutations aléatoires visant à réduire les corrélations entre les bits voisins. Finalement, le troisième type de décodeur (*avec mémoire*) est l'objet principal de ce mémoire, il tient en compte de la structure du modèle de bruit avec mémoire. Le cas des codes polaires (*cp*) et des codes polaires convolutifs (*cpc*) est comparé. La longueur de corrélation $\langle \ell_M \rangle$, définie à l'équation

$\langle \ell_M \rangle$	$q(B\|M)$	$q(M\|B)$	C_{12}
2.5	0.4	0.08	0.507
4	0.25	0.05	0.602
7	0.145	0.029	0.696
13	0.075	0.015	0.780
20	0.05	0.01	0.817
40	0.025	0.005	0.861

TABLE 4.1 Différents paramètres pour les canaux étudiés dans la figure 4.1 a), c) et d) où le ratio $\rho = 5$.

2.33, est utilisée comme paramètre d'intérêt pour l'analyse. La figure 4.1 présente les résultats des simulations de performance des codes et décodeurs considérés en fonction de divers paramètres du modèle de bruit de Gilbert-Elliott. Pour l'ensemble de ces résultats, on fixe le ratio $\rho = 5$ et $h = 0.9$, ce qui donne une probabilité d'erreur moyenne $P_{moy} \approx 0.15$. La table 4.1 présente les divers paramètres pour $q(B|M)$ et $q(M|B)$ avec la longueur $\langle \ell_M \rangle$ correspondante.

4.3 Analyse

L'ensemble des résultats de la figure 4.1 indique une meilleure performance dans le cas des codes polaires convolutifs que le cas des codes polaires pour les paramètres du modèle de Gilbert-Elliott considérés. Ceci vient donc étendre les résultats de [16], [18] et [32] quant à l'avantage des codes polaires convolutifs. La figure 4.1 a) montre que dans le régime de bruit donné, l'ajout d'un algorithme d'entrelacement pour réduire les corrélations aux bits voisins n'a peu ou pas d'effet sur la performance des codes. En effet, remarquons que les courbes avec le suffix *permutation* sont au même niveau que celles de l'algorithme standard. Une des raisons possibles pour cette performance médiocre de l'algorithme d'entrelacement est reliée au fait que la probabilité d'erreurs moyenne du canal est de $P_{moy} = P(B)h = 0.15$. Effectivement, après l'application de permutations aléatoires le modèle de bruit avec mémoire

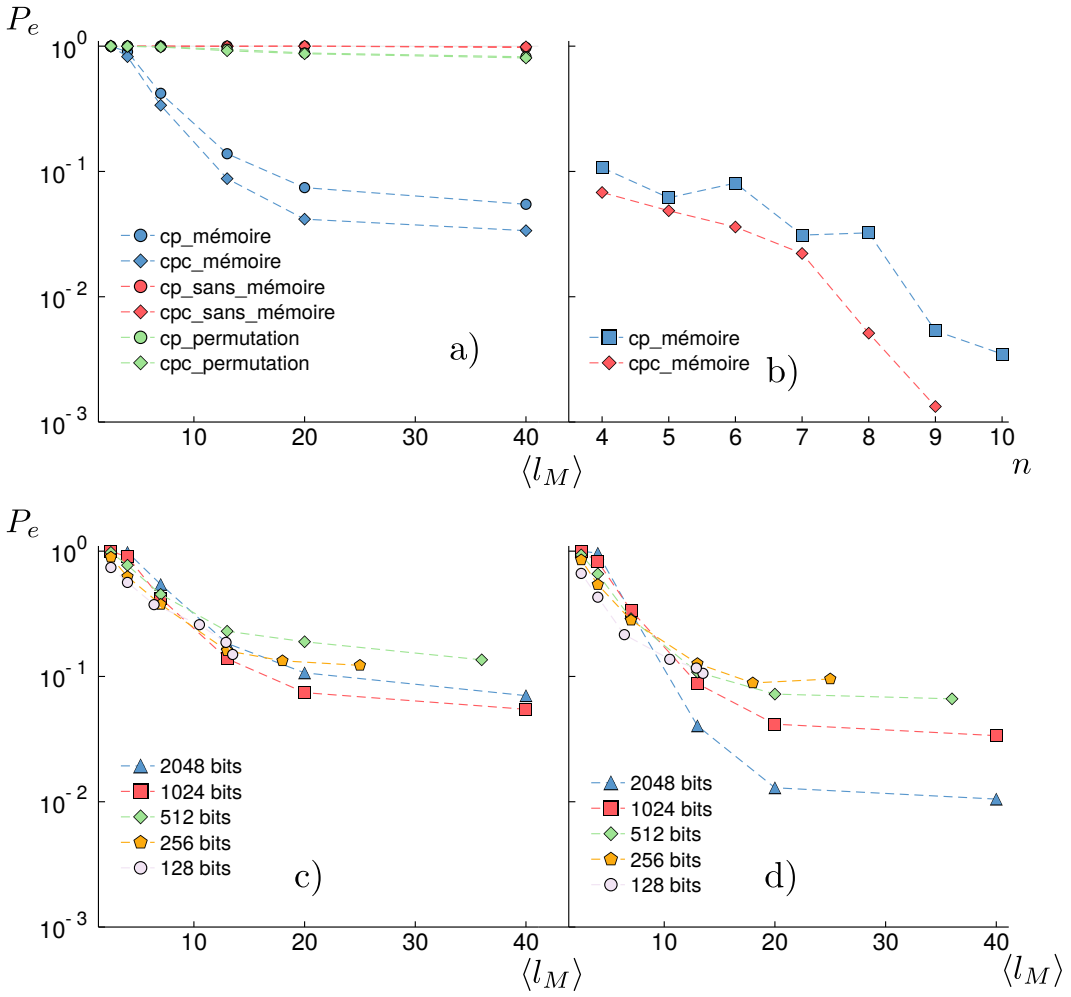


FIGURE 4.1 Taux d'erreurs par trame en fonction de différents codes, décodeurs et paramètres du canal de Gilbert-Elliott. Les figures a), c) et d) sont obtenues en fixant $h = 0.9$ et $\rho = 5$ et en faisant varier la longueur moyenne de la rafale en utilisant des codes de tailles $N = 2^{10}$ et un rendement $R = \frac{1}{2}$. Les paramètres exacts sont données dans la table 4.1. En a), il s'agit d'une comparaison entre les différents types de décodeurs et codes. Les cercles correspondent aux codes polaires alors que les losanges correspondent aux codes polaires convolutifs. En bleu, il s'agit du décodeur développé dans ce mémoire (adapté au bruit avec mémoire), en rouge il s'agit de la construction standard prenant en compte la probabilité d'erreur moyenne et en vert il s'agit d'une construction standard avec la technique d'entrelacement (permutation). En b), les codes polaires (carré bleu) et les codes polaires convolutifs (losange rouge) sont comparés avec l'algorithme de décodage proposé en fonction du niveau de polarisation n pour un rendement $R = \frac{1}{3}$ avec les paramètres suivants : $h = 0.9$, $q(B|M) = 0.05$ et $q(M|B) = 0.01$. Les figures c) et d) étudient la performance du décodeur proposé selon différents niveaux de polarisation pour les codes polaires en c) et les codes polaires convolutifs en d).

peut être vu comme une série de canaux binaires symétriques indépendants avec probabilité d'erreur ≈ 0.15. Selon les résultats de [16] et [18] une probabilité d'erreur physique de 0.15 est relativement élevée pour un code avec rendement $R = \frac{1}{2}$. Il n'est donc pas surprenant d'obtenir un résultat de la sorte. D'autre part, remarquons aussi que dans la limite où $\langle \ell_M \rangle \to 1$,les performances sont similaires pour chacun des décodeurs dans le cas des figures 4.1 a), c) et d). En effet, cette limite correspond à un cas où le modèle de bruit tend vers un canal sans mémoire puisque $q(B|M)$ et $q(M|B)$ deviennent assez grands. Ainsi, comme le décodeur proposé dans ce mémoire est une généralisation du décodeur standard, il est attendu que toutes les courbes de performance se rencontrent en ce point.

La figure 4.1 b), semble indiquer que les codes polaires convolutifs polarisent plus rapidement les canaux avec mémoire que les codes polaires. Dans les deux cas, une diminution de la probabilité d'erreur après décodage avec la taille du système est notable, il s'agit d'une signature d'un code qui atteint la capacité. Finalement, les résultats de la figure 4.1 c) et d) indiquent encore une meilleure performance pour les codes polaires convolutifs (d)). L'existence d'une valeur seuil est aussi observée dans le cas des codes polaires convolutifs près de $\langle \ell_M \rangle = 10$. En deçà de cette valeur seuil, la performance du code dégrade avec la taille du système alors que dépassée cette valeur, une diminution de la probabilité d'erreur après décodage en fonction de la taille du système est observée. L'existence de cette valeur seuil dans le cas des codes polaires convolutifs appuie l'hypothèse que ces codes peuvent atteindre la capacité pour les paramètres du modèle de bruit considérés. Dans le cas des codes polaires, aucune valeur seuil n'est observée, ceci peut être dû aux effets de tailles finis.

Conclusion

En conclusion, l'aspect fondamental de ce mémoire est l'usage des réseaux de tenseurs dans un problème de décodage. En effet, les réseaux de tenseurs s'avèrent être de puissants outils applicables en théorie des codes. Dans le cadre de ce projet, ces outils ont permis de généraliser le décodeur par annulation successive pour prendre en compte une structure de bruit avec mémoire. Cette structure de bruit avec mémoire peut simplement se décrire comme un réseau de tenseurs ayant la topologie d'une chaîne.

À la lumière des résultats obtenus, l'usage d'un décodeur utilisant la structure du bruit avec mémoire performe toujours mieux qu'un décodeur ayant seulement une propriété globale telle que la probabilité d'erreur moyenne. De plus, pour les canaux considérés, les codes polaires convolutifs donnent une meilleure performance que les codes polaires standards.

Une direction intéressante pour poursuivre ce plan de recherche pourrait être l'étude numérique des codes polaires et codes polaires convolutifs soumis au canal d'interférence intersymbole. L'algorithme développé pourrait simplement être adapté. Une autre direction intéressante consiste à généraliser les méthodes présentées dans ce mémoire au cas quantique.

Annexe A

Chaînes de Markov

Définition 16 *Soit $\mathcal{P}$ un processus stochastique définit par $X_0, X_1, X_2, ...$ une suite de variable aléatoire chacune à valeur dans l'ensemble $S = \{0, 1, ..., M\}$. Alors, $\mathcal{P}$ est une chaîne de Markov si $q(X_{n+1} = j|X_n = i, X_{n-1} = i_{n-1}, ..., X_0 = i_0) = q(X_{n+1} = j|X_n = i)$.*

L'ensemble S peut être interprété comme l'état d'un système au cours du temps de sorte que $X_n = i$ indique que l'état du système au temps n est i. La probabilité d'une transition au temps $n + 1$ sachant les états au temps inférieur est donc une quantité d'intérêt. La caractéristique d'une chaîne de Markov est que cette probabilité de transition dépend seulement de l'état au temps n. La probabilité conjointe d'une chaîne de Markov se factorise de la manière suivante :

$$q(x_n, x_{n-1}, ..., x_0) = q(x_0)q(x_1|x_0)q(x_2|x_1)...q(x_n|x_{n-1}). \tag{A.1}$$

Définition 17 *Une chaîne de Markov est indépendante dans le temps si $q(x_{n+1}|x_n)$ ne dépend pas de n.*

Une chaîne de Markov indépendante du temps est caractérisée entièrement par un vecteur décrivant l'état initial $\vec{\nu}_0$ et une matrice de transition de probabilité $q = [q_{ij}] \; \forall \, i, j \in \{1, 2, ..., m\}$. L'état initial est représenté par un vecteur de dimension

m et il contient la distribution sur les états au temps 0. Soit ν_0 l'état initial, alors $\vec{\nu}_1 = q^T \vec{\nu}_0$. De manière plus générale,

$$\vec{\nu}_n = (q^T)^n \vec{\nu}_0. \tag{A.2}$$

Ainsi, la matrice de probabilité de transition peut être vu comme un opérateur d'évolution temporelle sur un état.

Définition 18 *Un processus stochastique est appelé stationnaire si*

$$q(X_1 = x_1, X_2 = x_2, ..., X_n = x_n) = q(X_{1+l} = x_1, X_{2+l} = x_2, ..., X_{n+l} = x_n) \tag{A.3}$$

pour tout temps n, translation dans le temps l et $x_1, x_2, ..., x_n \in S$.

Définition 19 *Une distribution sur les états $\vec{\nu}_n$ est une distribution stationnaire si*

$$\vec{\nu}_n = q^T \vec{\nu}_n. \tag{A.4}$$

Selon ces définitions, la distribution stationnaire est un vecteur propre de valeur propre $+1$ de la matrice de transition. L'existence et l'unicité de cette distribution stationnaire provient des concepts d'irréductibilité et d'apériodicité d'une chaîne de Markov.

Définition 20 *Une chaîne de Markov est irréductible s'il est possible d'effectuer une transition de n'importe quel état vers tous les autres états en un temps fini.*

Une chaîne de Markov indépendante du temps peut être représentée par un graphe de transition qui est équivalent à la matrice de transition. Soit une matrice de transition pour une chaîne de Markov à m états, alors le graphe de transition correspondant contiendra m noeuds représentant chacun des états. De plus, pour chaque élément non nul de la matrice P_{ij}, une flèche relie le noeud de l'état i au noeud de l'état j. La figure A.1 présente un exemple de graphe de transition décrivant la

matrice de transition suivante,

$$q = \begin{pmatrix} 0 & 0 & q_{13} \\ q_{21} & 0 & q_{23} \\ 0 & q_{32} & q_{33} \end{pmatrix}. \tag{A.5}$$

Dans cet exemple, il est possible de parcourir un chemin dans ce graphe pour passer d'un état à l'autre. Le nombre de chemins emprunté donne la valeur du pas de temps nécessaire. Suivant l'exemple de la figure A.1, si l'état initial est S_1, alors le temps minimal requis pour rejoindre l'état S_2 est de 2. Par cette méthode de comptage, il est possible de définir la *période* d'un état S_i comme étant le plus grand commun diviseur de l'ensemble des temps requis pour revenir à l'état S_i. Si la période est de 1, alors l'état est *apériodique*. Une chaîne de Markov avec tous ses états apériodiques est nommée une chaîne de Markov apériodique.

Définition 21 *Soit $\mathcal{P}$ une chaîne de Markov indépendante du temps et à états finis. Si $\mathcal{P}$ est irréductible et apériodique, alors la chaîne de Markov est ergodique.*

Définition 22 *Soit $\mathcal{P}$ une chaîne de Markov ergodique, alors il existe une unique distribution stationnaire $\vec{\nu}_s$ et pour toute distribution $\vec{\nu}_0$,*

$$\lim_{n\to\infty} (q^T)^n \vec{\nu}_0 = \vec{\nu}_s. \tag{A.6}$$

La chaîne de Markov de la figure 2.5 est irréductible et apériodique, donc ergodique. Il existe donc un unique état stationnaire qui satisfait $q^T\vec{\nu} = \vec{\nu}$. Avec comme matrice de transition

$$q = \begin{pmatrix} q(B|B) & q(B|M) \\ q(M|B) & q(M|M) \end{pmatrix} \tag{A.7}$$

et l'état stationnaire

$$\vec{\nu} = \begin{pmatrix} p_B \\ p_M \end{pmatrix}. \tag{A.8}$$

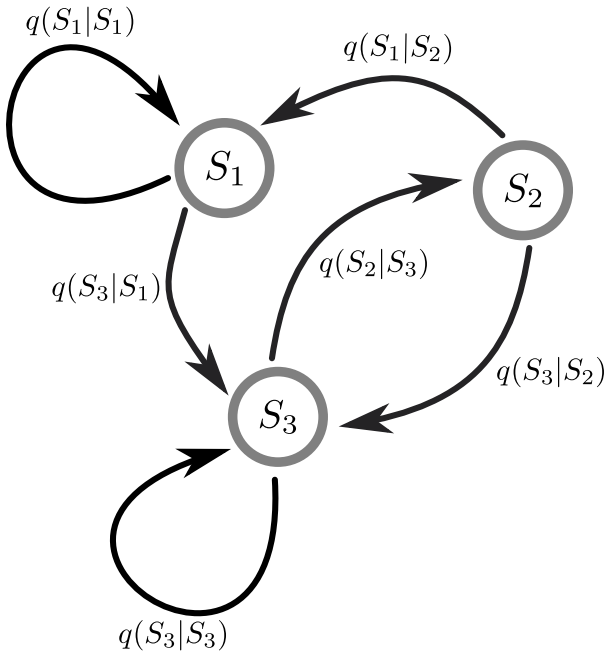


FIGURE A.1 Exemple d'un diagramme représentant un processus de Markov à 3 états.

Il suffit donc de résoudre l'équation

$$\begin{pmatrix} q(B|B) & q(B|M) \\ q(M|B) & q(M|M) \end{pmatrix}^T \begin{pmatrix} p_B \\ p_M \end{pmatrix} = \begin{pmatrix} p_B \\ p_M \end{pmatrix}. \tag{A.9}$$

Ce qui donne,

$$\begin{pmatrix} p_B \\ p_M \end{pmatrix} = \frac{1}{q(M|B) + q(B|M)} \begin{pmatrix} q(B|M) \\ q(M|B) \end{pmatrix}. \tag{A.10}$$

Annexe B

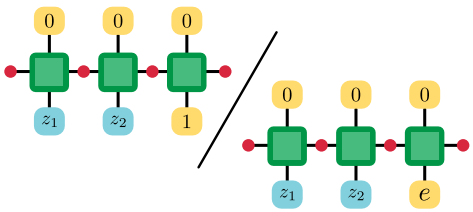
Calcul de la capacité du canal de Gilbert-Elliott

Le calcul de la capacité du modèle de Gilbert-Elliott peut être effectué grâce à la formule 2.34. On propose d'utiliser les réseaux de tenseurs pour effectuer le calcul de la capacité. À titre d'exemple, les figures présentées sont pour $n = 3$, mais on peut facilement généraliser pour n plus grand.

Il faut donc calculer le terme Q_n, observons la relation suivante :

$$Q_n = P(z_n = 1 | z_1^{n-1}, s_0) = \frac{P(z_n = 1, z_1^{n-1} | s_0)}{P(z_1^{n-1} | s_0)} \tag{B.1}$$

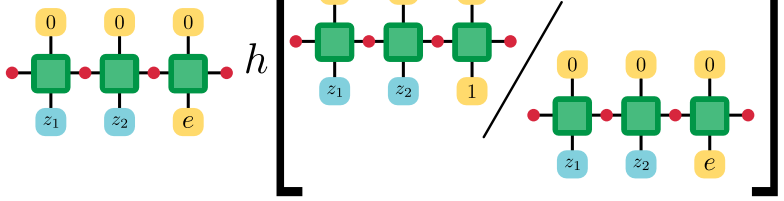
En termes de réseaux de tenseurs,

$Q_3 =$ 

(B.2)

La valeur moyenne de l'entropie binaire de Q_3 est donc donnée par,

$$\mathbb{E}[h(Q_3)] = \sum_{z_1 z_2} \; h\left[\; \Big/ \; \right] \tag{B.3}$$



Pour les capacités de la table 4.1, la valeur $n = 12$ a été utilisé.

Bibliographie

[1] Itai Arad and Zeph Landau. Quantum computation and the evaluation of tensor networks. *arXiv:0805.0040 [quant-ph]*, April 2008. arXiv : 0805.0040.

[2] E. Arikan. Source polarization. In *2010 IEEE International Symposium on Information Theory*, pages 899–903, June 2010.

[3] Erdal Arikan. Channel polarization : A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on Information Theory*, 55(7) :3051–3073, July 2009. arXiv : 0807.3917.

[4] Dieter M. Arnold. *Computing information rates of finite-state models with application to magnetic recording*. PhD thesis, ETH Zurich, 2003.

[5] Claude Berrou, editor. *Codes and turbo codes*. Collection IRIS. Springer-Verlag, Paris, 2010.

[6] Valerio Bioglio, Carlo Condo, and Ingmar Land. Design of Polar Codes in 5g New Radio. *arXiv:1804.04389 [cs, math]*, April 2018. arXiv : 1804.04389.

[7] David Blackwell, Leo Breiman, and A. J. Thomasian. Proof of Shannon's Transmission Theorem for Finite-State Indecomposable Channels. *The Annals of Mathematical Statistics*, 29(4) :1209–1220, 1958.

[8] Benjamin Bourassa, Maxime Tremblay, and David Poulin. Convolutional Polar Codes on Channels with Memory. *arXiv:1805.09378 [cs, math]*, May 2018. arXiv : 1805.09378.

[9] Sergey Bravyi, Martin Suchara, and Alexander Vargo. Efficient algorithms for maximum likelihood decoding in the surface code. *Phys. Rev. A*, 90 :032326, Sep 2014.

[10] Jacob C. Bridgeman and Christopher T. Chubb. Hand-waving and Interpretive Dance : An Introductory Course on Tensor Networks. *Journal of Physics A: Mathematical and Theoretical*, 50(22) :223001, June 2017. arXiv : 1603.03039.

[11] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory 2nd Edition*. Wiley-Interscience, Hoboken, N.J, 2 edition edition, July 2006.

[12] Andrew S. Darmawan and David Poulin. Tensor-network simulations of the surface code under realistic noise. *Phys. Rev. Lett.*, 119 :040502, Jul 2017.

[13] Andrew S. Darmawan and David Poulin. Linear-time general decoding algorithm for the surface code. *Phys. Rev. E*, 97 :051302, May 2018.

[14] E. O. Elliott. Estimates of error rates for codes on burst-noise channels. *The Bell System Technical Journal*, 42(5) :1977–1997, September 1963.

[15] Glen Evenbly and Guifre Vidal. A class of highly entangled many-body states that can be efficiently simulated. *Physical Review Letters*, 112(24), June 2014. arXiv : 1210.1895.

[16] Andrew J. Ferris and David Poulin. Branching MERA codes : a natural extension of polar codes. *arXiv:1312.4575 [quant-ph]*, December 2013. arXiv : 1312.4575.

[17] Andrew J. Ferris and David Poulin. Tensor Networks and Quantum Error Correction. *Physical Review Letters*, 113(3), July 2014.

[18] Andrew James Ferris, Christoph Hirche, and David Poulin. Convolutional Polar Codes. *arXiv:1704.00715 [cs, math]*, April 2017. arXiv : 1704.00715.

[19] R. G. Gallager. *Information Theory and Reliable Communication*. Wiley, New York, 1968.

[20] E. N. Gilbert. Capacity of a burst-noise channel. *The Bell System Technical Journal*, 39(5) :1253–1265, September 1960.

[21] A. J. Goldsmith and P. P. Varaiya. Capacity, mutual information, and coding for finite-state Markov channels. *IEEE Transactions on Information Theory*, 42(3) :868–886, May 1996.

[22] R. W. Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2) :147–160, April 1950.

[23] Shu Lin and Daniel J. Costello. *Error Control Coding: Fundamentals and Applications*. Pearson-Prentice Hall, 2004. Google-Books-ID : 0nVfQgAACAAJ.

[24] Florence Jessie MacWilliams and Neil James Alexander Sloane. *The Theory of Error-correcting Codes*. Elsevier, 1977. Google-Books-ID : nv6WCJgcjxcC.

[25] Igor L. Markov and Yaoyun Shi. Simulating quantum computation by contracting tensor networks. *SIAM Journal on Computing*, 38(3) :963–981, January 2008. arXiv : quant-ph/0511069.

[26] R. Morozov and P. Trifonov. Efficient sc decoding of convolutional polar codes. In *Proceedings of International Symposium on Information Theory and Applications*, 2018.

[27] M. Mushkin and I. Bar-David. Capacity and coding for the gilbert-elliott channels. *IEEE Transactions on Information Theory*, 35(6) :1277–1290, November 1989.

[28] Roman Orus. A Practical Introduction to Tensor Networks : Matrix Product States and Projected Entangled Pair States. *Annals of Physics*, 349 :117–158, October 2014. arXiv : 1306.2164.

[29] Fernando Pastawski, Beni Yoshida, Daniel Harlow, and John Preskill. Holographic quantum error-correcting codes : toy models for the bulk/boundary correspondence. *Journal of High Energy Physics*, 2015(6) :149, Jun 2015.

[30] B Pirvu, V Murg, J I Cirac, and F Verstraete. Matrix product operator representations. *New Journal of Physics*, 12(2) :025012, 2010.

[31] David Poulin, Angie Qarry, Rolando Somma, and Frank Verstraete. Quantum Simulation of Time-Dependent Hamiltonians and the Convenient Illusion of Hilbert Space. *Physical Review Letters*, 106(17), April 2011.

[32] Tobias Prinz and Peihong Yuan. Successive Cancellation List Decoding of BMERA Codes with Application to Higher-Order Modulation. *arXiv:1807.03601 [cs, math]*, July 2018. arXiv : 1807.03601.

[33] M. Rezaeian. Computation of Capacity for Gilbert-Elliott Channels, Using a Statistical Method. In *2005 Australian Communications Theory Workshop*, pages 56–61, Brisbane, Australia, 2005. IEEE.

[34] Michael Rice, Jeffrey Slack, Brian Humpherys, and Deborah S. Pinck. K-Band Land-Mobile Satellite Channel Characterization Using Acts. *International Journal of Satellite Communications*, 14(3) :283–296, May 1996.

[35] Eren Sasoglu and Ido Tal. Polar Coding for Processes with Memory. *arXiv:1602.01870 [cs, math]*, February 2016. arXiv : 1602.01870.

[36] C E SHANNON. A Mathematical Theory of Communication. page 55.

[37] Yaoyun Shi, Luming Duan, and Guifre Vidal. Classical simulation of quantum many-body systems with a tree tensor network. *Physical Review A*, 74(2), August 2006. arXiv : quant-ph/0511070.

[38] E. Miles Stoudenmire and David J. Schwab. Supervised Learning with Quantum-Inspired Tensor Networks. *arXiv:1605.05775 [cond-mat, stat]*, May 2016. arXiv : 1605.05775.

[39] Maxime Tremblay, Benjamin Bourassa, and David Poulin. Depth versus Breadth in Convolutional Polar Codes. *arXiv:1805.09306 [cs, math]*, May 2018. arXiv : 1805.09306.

[40] R. Wang, J. Honda, H. Yamamoto, R. Liu, and Y. Hou. Construction of polar codes for channels with memory. In *2015 IEEE Information Theory Workshop - Fall (ITW)*, pages 187–191, October 2015.

[41] J. R. Yee and E. J. Weldon. Evaluation of the performance of error-correcting codes on a Gilbert channel. *IEEE Transactions on Communications*, 43(8) :2316–2323, August 1995.

[42] Yun Q.Shi, Xi Min Zhang, Zhi-Cheng Ni, and N. Ansari. Interleaving for combating bursts of errors. *IEEE Circuits and Systems Magazine*, 4(1) :29–42, 2004.

[43] E. Şaşoğlu. Polarization in the presence of memory. In *2011 IEEE International Symposium on Information Theory Proceedings*, pages 189–193, July 2011.

[44] Eren Şaşoğlu. Polarization and Polar Codes. *Foundations and Trends® in Communications and Information Theory*, 8(4) :259–381, 2011.